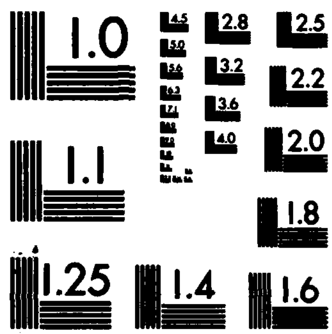
AD-A140 149 EMPIRICAL INVESTIGATION OF AIDS FOR NON-PROGRAMMING USERS IN DEVELOPING C..(U) PERFORMANCE MEASUREMENT ASSOCIATES INC VIENNA VA E M CONNELLY MAR 84 1/2

UNCLASSIFIED PMA-84-822 N00014-82-C-0499 F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

TECHNICAL
REPORT

EMPIRICAL INVESTIGATION OF AIDS FOR NON-PROGRAMMING USERS IN DEVELOPING COST-EFFECTIVE REQUIREMENTS SPECIFICATIONS

PMA

12

TECHNICAL
REPORT

Edward M. Connelly

EMPIRICAL INVESTIGATION OF AIDS FOR NON-PROGRAMMING USERS IN DEVELOPING COST-EFFECTIVE REQUIREMENTS SPECIFICATIONS

This research is supported
by Engineering Psychology
Group, Office of Naval Research

March 1984

PMA
410 Pine Street, S.E.
Suite 300
Vienna, Virginia 22180
(703) 938-1600

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 84-822	2. GOVT ACCESSION NO. AD-A140149	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Empirical Investigation of Aids for Non-Programming Users in Developing Cost-Effective Requirements Specifications		5. TYPE OF REPORT & PERIOD COVERED Technical Report July 1982 thru October 1983
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Edward M. Connelly		8. CONTRACT OR GRANT NUMBER(s) N00014-82-C-0499
9. PERFORMING ORGANIZATION NAME AND ADDRESS Performance Measurement Associates, Inc. 410 Pine Street, S.E. Suite 300 Vienna, Virginia 22180		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61153N 42; RR04209; RR0420901; NR 196-175
11. CONTROLLING OFFICE NAME AND ADDRESS Engineering Psychology Group Office of Naval Research 800 N. Quincy St. Arlington, VA 22217		12. REPORT DATE March 1984
		13. NUMBER OF PAGES 104
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Department of the Navy Office of Naval Research Arlington, VA 22217		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Software Engineering, Software Experiments, Requirements Specifications Software Human Factors, Cost-Effective Specifications, Non-Programming Users

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The difficulties of the non-programming user in developing quality software-requirements specifications are well known and described in the literature. Typically, the user will work with a software expert to develop the specifications. During this process the user must learn new terms and concepts, and must attempt to identify the required functions of the resulting software product. Further, the user must learn the

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

relative cost of the various candidate features of the product. Finally, the user must communicate, verbally or in writing, the requirements specifications in clear, unambiguous language.

The goal of this investigation was to evaluate the capability of non-programming users (in this case experienced inventory managers) to develop specifications for an inventory error-control system. Participants specified tests from a set of available tests to detect possible errors in inventory change-records, i.e., in inventory updates. Participants worked on problems at various levels of complexity. For each set of change-record tests specified, which defined a candidate design, the total system cost was automatically calculated and fed back to the participants. Participants attempted to specify the least-cost design.

Costing-aids were provided that were analogs of aids that are expected to be presently available. The costing-aids provided an increasing data on system cost: the total system cost, the total system cost plus the costs of each part of the system, and an automatic sort of previous designs according to cost. Problem-complexity had a strong effect on performance; greater problem-complexity resulted in more costly designs. Further, the more complete costing-aids tended to degrade performance on the simple problems, yet they improved performance on the more complex problem. Those effects, however, were not statistically significant.

It was concluded that the ability of non-programming users to develop least-cost requirements specifications was poor when using any of the aids that were made available. The experimental results, however, suggested new approaches for procedural and computational support for the non-programming user, that could lead to more suitable costing aids.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTD)

TABLE OF CONTENTS (CONTD)

LIST OF FIGURES

LIST OF TABLES

LIST OF TABLES (CONTD)

Abstract

The difficulties of the non-programming user in developing quality software-requirements specifications are well known and described in the literature. Typically, the user will work with a software expert to develop the specifications. During this process the user must learn new terms and concepts, and must attempt to identify the required functions of the resulting software product. Further, the user must learn the relative cost of the various candidate features of the product. Finally, the user must communicate, verbally or in writing, the requirements specifications in clear, unambiguous language.

The goal of this investigation was to evaluate the capability of non-programming users (in this case experienced inventory managers) to develop specifications for an inventory error-control system. Participants specified tests from a set of available tests to detect possible errors in inventory change-records, i.e., in inventory updates. Participants worked on problems at various levels of complexity. For each set of change-record tests specified, which defined a candidate design, the total system cost was automatically calculated and fed back to the participants. Participants attempted to specify the least-cost design.

Costing-aids were provided that were analogs of aids that are expected to be presently available. The costing-aids provided an increasing data on system cost: the total system cost, the total system cost plus the costs of each part of the system, and an automatic sort of previous designs according to cost. Problem-complexity had a strong effect on performance; greater problem-complexity resulted in more costly designs. Further, the more complete costing-aids tended to degrade performance on the simple problems, yet they improved performance on the more complex problem. Those effects, however, were not statistically significant.

It was concluded that the ability of non-programming users to develop least-cost requirements specifications was poor when using any of the aids that were made available. The experimental results, however, suggested new approaches for procedural and computational support for the non-programming user, that could lead to more suitable costing aids.

INTRODUCTION

Problem Statement

Although errors in software occur in all stages of the software life cycle, errors that occur early in the cycle, especially in the preparation of software-requirements specifications (RS), are critical because they are difficult to detect. Also, the cost of correcting a requirement-specification error increases as its detection is delayed through subsequent software development states. Further complications, according to Boehm (1976), are that most errors in software-development occur in the early design stages and are frequently errors of omission.

Incomplete or inaccurate requirements specifications cause difficulty in other software development stages as well: systematic, top-down design suffers from a lack of complete specifications; testing is inadequate due to lack of complete, accurate requirements to test against; and project management suffers from the lack of a complete statement against which progress can be measured.

Many existing methodologies developed to improve software quality unfortunately apply only to the design stages (both preliminary and detailed) and the subsequent stages. These methodologies, which are directed toward improving the software-development process only after RS have been specified, as a result neglect the process of producing quality RS, themselves. This imbalance in the distribution of software methodologies and aids, in which software design and testing are favored over the preparation of RS, is illustrated by the lack of a requirement-preparation "block" in typical descriptions of the software development process, as shown in Figure 1. The process as typically visualized begins with a requirements block, as if to indicate that the process starts with RS already in existance. Improvements in the requirements-development process have been generally limited to the development of notational methods for recording requirements. Actually, of course, the process starts, as shown in Figure 2, with user-needs, which may or may not be well understood by the user, but from which the RS must be developed. It is this actual process, the transformation of vague user-needs into precise RS, often with the user and a software expert working together, that is of interest here.

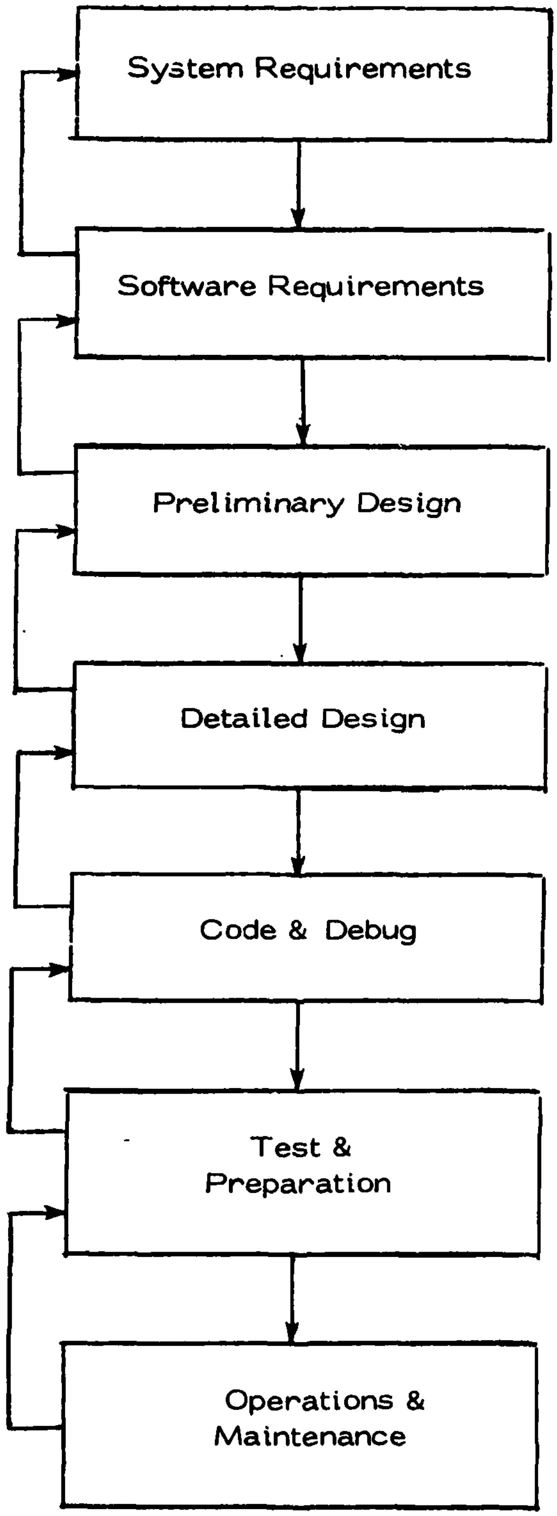


Figure 1. Typically but Erroneously Assumed Software Life Cycle

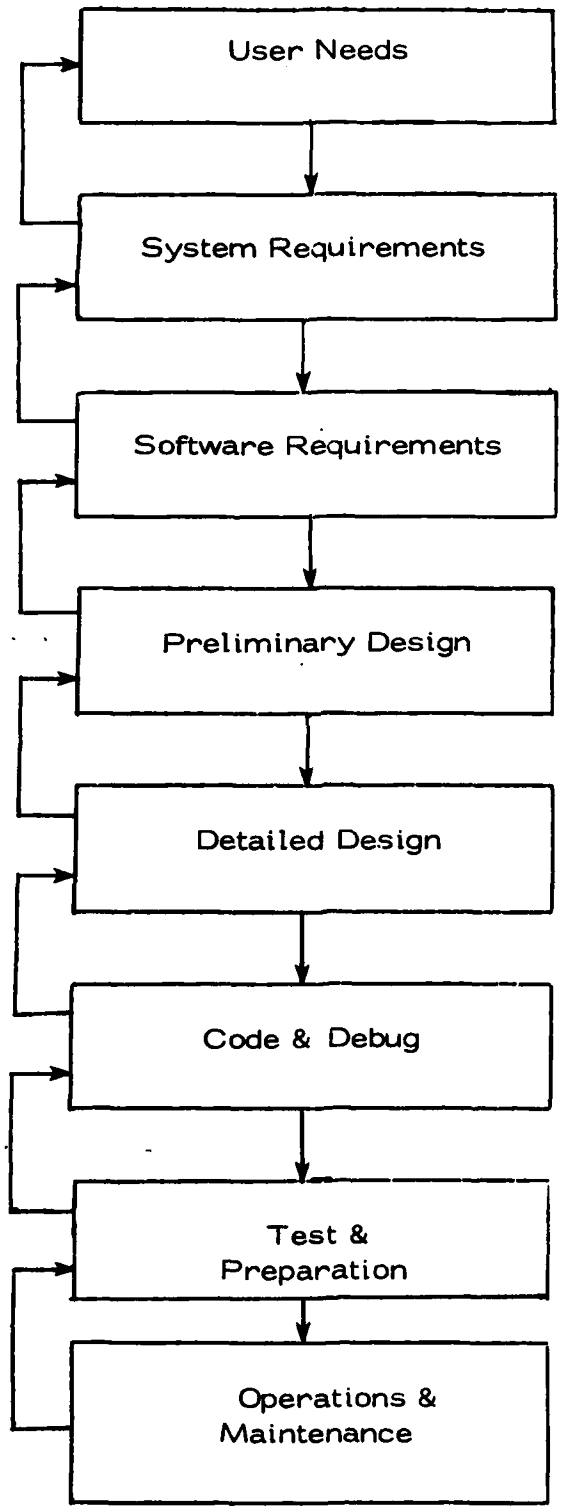


Figure 2. Actual Software Life Cycle

Background

At present, the tools available for software development fall into one of two classes. The first includes the well-known design-aids such as Structured Design (Yourdon & Constantine, 1979), Jackson's Method (Jackson, 1975), and Logical Construction of Program (Warnier, 1981 and Orr, 1981). All of these design aids use RS, which are assumed to be correct, as their starting point.

The second class of aids provides structures with which to record and analyze RS. In this latter class is a system called ISDOS (Teichroew & Hershey, 1979), which uses a problem-statement language (PSL) and a problem-statement analyser (PSA). ISDOS permits a formal description of a system in terms of entities, classes, and relationships, and automatically provides analytical summaries, such as problem-statements, directories, hierarchical-structure reports, and graphical summaries of data flow and data relationships. Another example in the class is a method called Software Requirements Engineering Programs (SREP), described by Boehm (1976) in a survey of methodologies. SREP uses the data-management system of ISDOS and, in addition, produces functional simulations from requirements statements. SREP is used for configuration control, traceability from requirements to design, and report generation. Still a further method in the second class, Structured Analysis for Requirements Definition, is described by Ross and Schoman (1977). Part of it is a Structured Analysis and Design Technique (SADT) for analyzing requirements using graphical techniques.

All these methods, despite their complexity, contain a common limitation: that of providing a structure for recording RS and then for analyzing those RS, but not for supporting the process of developing RS from a user's needs.

In an extensive survey and review of the status of software-requirements methods, Ramamoorthy & So (1978) identified the same requirements-development problems referred to above; namely, that a large percentage of the total errors in software development occur in the requirements specifications, and that these errors cause serious problems leading to high costs, unresponsive products, slippage of production schedules, and difficulty in system operation and maintenance. They go on to briefly describe a number of methodologies for RS documentation which they feel may aid in the software-design process.

Richhart (1983) provides a list of the manual, semi-automated and automated tools available for recording RS and the related software-design processes. These tools, some of which have been identified above, are given in Appendix A. Richhart also gives a description of software-development procedures used in the Navy and the Air Force.

Desired Properties of Requirements Specifications

According to Howden (1982), requirements specifications are formal, contractual documents that define the system to be built. RS should thus be:

Complete
Precise and Unambiguous
Machine-Readable but Human Intelligible
Incrementally Constructed and Reviewed
Testable (Validable Plans)
Cross-Referenced
Change Controlled (Formal Reviews)
User Accepted

Problems with Requirements Specifications

In contrast to the desired features given by Howden (1982), Richhart (1983) developed a list of common problems with RS, given here in its entirety:

1. Developers and customers often interpret RS differently (different cultures).

2. Users do not know in detail what they want until they use a version of it.

3. Requirements change quickly and often frequently while the system is being developed: This includes both uncontrollable, external factors and internal redefinement.

4. Specification document is long and boring (not fully read or understood by users).

5. Specifications are frozen when users are only halfway up the learning curve.

6. The specification document itself contains errors. (In a typical case for a large corporation, 64% of its bugs were in requirements-analysis and design. (Martin,1982))

7. The act of providing what an end–user says he needs changes his perception of those needs. (Martin, 1982)

8. APD-systems staff usually take the initiative and design what they "thought the required system was."

9. Different needs for different users sometimes conflict.

10. Narrative specifications must be digested and converted into a useful, non-procedural model to serve as the basis for deriving the modules.

11. Specifications are seldom complete. Frequently are 20% incomplete. (Nolan, 1981)

12. The system the user gets may be exactly what he asked for, but it may not solve his problem at all. (Zahniser, 1981)

13. The eventual operational users will probably not be the same ones who helped develop the specs (due to personnel turnover, promotion, etc.), and the new users will usually want something different.

14. The fact is that many of the most important potential users of data processing do not know what they want until they experience using the system. When they first experience it, many changes are needed to make them comfortable with it and to meet their basic requirements. (Martin, 1982)

15. The requirements for management-information systems cannot be specified beforehand, and almost every attempt to do so has failed. The requirements change as soon as an executive starts to use his terminal. (Martin,1982)

16. The important problem is how to migrate from conventional programming and the traditional life cycle into the development of methodologies that are fast, flexible, interactive, and employable by end-users; methodologies in which interactive prototyping replaces formal, voluminous specifications which must be frozen; methodologies with which end-users can create and continuously modify their own applications. (Martin, 1982)

17. Approaches which use propositional calculus and formal-definition languages clearly do not relate to the grassroots user at all. (Zahniser, 1981)

Solutions Suggested for Solving RS Problems

Martin (1982)

1. High-level tools for user-driven specification and development.

2. Prototypes to largely replace the use of lengthly, written requirements documents. Application generators for prototyping. After prototyping, complete design to achieve machine efficiency, security, telecommunications networking, database creation, etc.

3. Use central data-base/information-control systems.

4. Highly interative/interrelated design & development.

Zahniser (1981)

1. Build a very high-level, abstract view of the system in user terms, employing:

   A. System surveys, (feasibility studies, baseline descriptions)
   B. Business requirements analysis.
   C. Structured problem definition.

2. Develop automated tools for working with the user, including:

   A. Interactive problem-analyser
   B. User-friendly data dictionary
   C. Cost-benefit analysis program
   D. A set of checklists, guidelines, and heuristics to serve as reference guides for measuring the completeness and quality of the design and requirements.

Bearley and Wood (1980)

The analyst must ... "assist the user in stating these perceived problems in a clear and understandable manner. To do this required a form of problem analysis that not only identifies the user's perceived problems, but also quantifies reasons for the problems and defines criteria by which the user, analyst and management can determine whether the problem has been resolved."

Research on Development of Requirements Specifications

Miller (1978) investigated the interactive process between a user (client) and software designer in analyzing the user's needs, establishing RS, and developing a software design. His description of the interactive process identified four steps, and is presented below. For ease of reference in what follows, we give all four of Miller's steps, even though our interest here is limited to the first two steps.

The four steps Miller identified are:

1. Problem understanding, arriving at a general agreement as to what are:

   a. the goal objectives,
   b. the system or environments involved,
   c. the constraints (on performance, delivery, costs, etc.),
   d. the resources available for system-design development.

2. Functional requirements specifications determining precisely what the final product must be like, including:

   a. every important aspect of the product's internal performance,
   b. the characteristics of its embedded operator/user population,
   c. its relationship to other systems and environments, and
   d. the development constraints

3. Overall high-level design translating the functional requirements into a comprehensive design which specifies the major components of the to-be-developed product, and describing for each component:

   a. the goals to be achieved by the component,
   b. the characteristics of all factors to which the component is to be sensitive, i.e., the input,
   c. the characteristics of the effects the component must achieve, i.e., the output,
   d. the internal structures of the component, and
   e. the general principle(s) of any operation sequences within the component information-processing procedures.

4. Detailed design suitable for prototype development.

The steps of interest here are the first and second, which start with the initial discussions of the problem with the client and end with preparation of formal RS. We will not treat the high-level or detailed-design steps.

Miller investigated the transformation of a client's vague, initial specifications into precise and formal specifications. He described, in particular, the functions of the client and the software designer by describing the interchange between the two, and he noted that designers often use the technique of suggesting particular pieces of equipment or procedures that might be (or might at least approximate) an acceptable solution. The client, in rejecting some of these suggestions, modifies his own requirements statements. As a result of this designer-client interchange, the client clarifies his own understanding of the problem, and, by working together, the client and designer arrive at an acceptable solution.

Miller pointed out that the role of the designer is to provide facts about the real world, in terms of the properties of equipment and alternative solutions, as well as to ask questions which, while providing clarification, frequently may have the effect of inducing the client to identify a new problem or to better conceptualize the present problem. Miller further identified a sequence of six states which the client and designer use sequentially. These six states are:

1. Goal statement
2. Goal elaboration
3. (Sub) Solution outline
4. (Sub) Solution elaboration
5. (Sub) Solution explication
6. Agreement on (Sub) solution.

Miller indicated that this state-sequence was used iteratively, but that sometimes the sequence was truncated in order to start a new sequence in the pursuit of a different solution.

The results by Miller suggest that the process of transforming a user's needs into a formal statement of requirements may benefit from the interchange between a client knowledgeable about his own needs and a software designer knowledgeable about the capabilities of computer systems. According to this model, the client's concept of his needs grows as a result of the interchange, and he or she becomes aware of new and different possible solutions to the problem. New solutions evolve iteratively until a final solution emerges that is accepted by both the client and designer as being complete and feasible.

The question arises, how comprehensive ought these interchanges to be to evolve a feasible solution? For instance, when preparing RS for a large computer system, it may not be possible for all the user-clients to have a useful interchange with one, or more, of the designers. Not only would there have to be multiple client-designer interchanges, but there would also have to be multiple interchanges (discussions of tradeoffs among the many interests) among the user-clients themselves, and perhaps multiple interchanges among the several designers. At present, when specifications for the development of a large software system are considered, the user-client develops formal specifications without extensive interchange concerning the ultimate designs. The RS are then presented to designers, perhaps in the form of a request-for-

quotation. Such a procedure, though often used for large software projects, is formal and prohibits the informal interchange described above that might well be used to advantage in the development of a smaller software system.

Goals of this Research Program

Goals of PMA's research program, for which this report is the first technical report, were to:

> Investigate the nature of the user/software-expert interaction in developing RS in order to identify factors that limit the quality of the resulting RS.
>
> Design aids to improve the quality of RS.
>
> Conduct experiments to test and evaluate the RS quality-improvement aids.

METHOD OF APPROACH

The Series of Experiments

A series of experiments was designed as a means for better understanding the process by which individuals not skilled in software, but skilled in at least one software-application area, develop RS by working with a software expert. A further purpose of the experiments was to design and test aids with which to assist software users in the process of developing RS. The four experiments which were planned sought to:

1. Identify the capabilities of, and the strategies employed by, software users in specifying a minimum-cost inventory-control system, and to establish a base-line study of presently available aids.

2. Develop and evaluate more sophisticated aids with which to assist a user in providing complete RS.

3. Develop and evaluate aids that a user and software expert could use together to build a working vocabulary of software specifics and application-specific terms.

4. Develop and evaluate aids to assist a neophyte user to develop RS.

This first technical report presents both a description and the results of the first experiment cited above -- an investigation of the capability of users to develop RS for a minimum-cost inventory-control system, including identification of the strategies employed by users who were successful in developing minimum-cost systems. Since the development of RS by an individual not skilled in software requires a dialogue with a software expert, several additional investigations are currently planned (See 2, 3, and 4 above) to evaluate the use of a computer to facilitate that dialogue.

Experiment Task: An Inventory-Control Problem

The experiment task was to develop software RS for an inventory-control problem by means of a recorded interaction between a user and a simulated software designer.

Problem: Inventory Control

An important aspect of inventory control is the maintenance of records: of present stock, of the amount of stock on order, of recent transactions, and of transaction histories. At periodic intervals, daily, weekly, or monthly, the old master inventory-file is read, transactions are recorded, new stock levels are computed, new stock-order recommendations and other reports are written, and a new, updated, master file is produced. These periodic updates of the old master file are accomplished by means of what are known as change-records. A change-record is the replacement of a datum in the old master file with a new datum. When all change-records have been entered, a new master file is produced.

If a change-record contains an error that is not detected prior to its incorporation in the new master file, the erroneous change-record may result in unnecessary costs to the organization. For instance, if a change-record error indicates that stock of a particular item is lower than the actual level, additional, unnecessary parts may be ordered. Conversely, if a change-record error indicates that the stock of a particular item is higher than the actual level, needed parts may not be ordered, possibly resulting in a production slow-down. Since change-record errors can result in unnecessary costs to the organization, it is well to consider specifying tests for these change-records in order to detect any errors before the change-records are entered into the new master file.

Tests of change-records, however, are not available without cost. Tests must be developed, evaluated, and programmed and maintained. Further, tests that result in "false alarms", i.e., indications of an error when one does not exist, result in unnecessary error-investigation costs. Expensive tests that only detect errors that have little cost-impact should not be used, and conversely, tests that detect errors with high cost-impact compared to the test cost should be used. Thus, change-record tests should be carefully specified so as to minimize the total cost to the organization. Total cost consists of the cost of undetected change-record errors plus the cost of test development and maintenance.

The Participant's Task

The participant's task was to specify a set of change-record tests that resulted in the least-total-cost system. Total cost was, as stated previously, the cost of undetected errors plus the cost (of development & maintenance) of the tests used.

Since the participant, who in all cases was a manager experienced in inventory systems, was not expected to have expert knowledge of the type and frequency of change-record errors, the cost-impact of each error, the cost of change-record tests, or the effectiveness of each test, a simulated software-expert was provided to calculate the expected cost of each of the participant's designs (the set of tests he or she specified) and to feed this total-cost information back to the participant.

Fourteen inventory records, shown in Table 1, were presented to the participant. Multiple types of errors, listed in Table 2, were allowed for each record, giving a total of 44 possible change-record/change-record error combinations. An error probability matrix, given in Appendix C, was developed containing the probability for each error-type for each change-record. The error probabilities selected were based on a search of the literature on error-rates for vaious types of tasks. It was found not to be possible, however, to use exact error-rates, for in many cases such error-rates were either unknown or were given in the literature as a broad range of values. Consequently, only representative error-probabilities were used.

The participant's task was to select none, one, or more than one test for each change-record. The maximum number of tests available, shown in Table 3, was 13; the number used for each problem was an experiment variable, described subsequently. Each test was supplied with a name and a brief description of its function. The description of a test's function was general and offered no more than guidance for the test's use. In some instances, the inappropriateness of a test was reasonably obvious. Thus, a numeric range test was inappropriate for the "part name" change-record because a name is not a number. Likewise, the "alpha range" test was inappropriate for checking the "part number", or "quantity on hand" change-records, etc., because quantities do not contain alphabetic characters. In general, although the test names and descriptions offered some guidance, reliable information was available only by trying a test in a specification and then observing the change in total cost for a change-record. When an additional test, for example test X, was specified for a change-record without altering the tests already specified for that change-record and the resultant cost decreased, reliable evidence was available that test X's use was beneficial.

The effectiveness of each change-record test was represented by a matrix of probabilities for correctly detecting each error-type for each change-record. Randomizing the effectiveness for the four problems (one pre-test problem and three experiment problems) required four probability matrices, given in Appendix C.

Table 1

Types of Records

Identification Records	
1.	Part Number
2.	Part Name
Records Indicating Present State	
3.	Quantity on Hand
4.	Quantity on Order
5.	Percent Damaged Received Damaged on Last Order
Expected Delivery Record	
6.	Quantity Expected in one Month
Records Indicating Conditions for Ordering	
7.	Delivery Time when Ordering Now
8.	Quantity Discount
9.	Recent Unit Price
Records of Historical Data	
10.	Quantity Used to Date This Year
11.	Quantity Used This Month
12.	Quantity Used Last Year
13.	Quantity Used Last Month
14.	Price Paid Last Year

Table 2

Types of Error Assigned to Each Change-Record

1. Part Number
   1. Wrong Part Number
   2. Interchange of Digits
   3. Digit Missing
   4. No Number

2. Part Name
   1. Wrong Part Name
   2. Misspelled Name
   3. No Name

3. Quantity on Hand
   1. Random Error
   2. Order not Recorded
   3. Stock Issue not Recorded

4. Quantity on Order
   1. Random Error
   2. Not Ordered
   3. Double Order

5. Percent Damage Last Order
   1. Random Error
   2. Not Recorded

6. Quantity Expected in One Month
   1. Random Error
   2. Quantity Not Updated Since Last Entry

7. Delivery Time When Ordering Now
   1. Random Error
   2. Time Not Updated Since Last Entry

8. Quantity Discount
   1. Random Error
   2. Discount Not Changed From Last Entry

Table 2 (Continued)

9. Unit Price
    1. Random Error
    2. Digits Interchanged
    3. Digit Missing

10. Quantity Used Last Year
    1. Random Error
    2. Quantity Not Changed Since Last Entry
    3. Digits Interchanged
    4. Digits Missing

11. Quantity Used This Month
    1. Random Error
    2. Quantity Not Changed Since Last Entry
    3. Digits Interchanged
    4. Digits Missing

12. Quantity Used Last Year
    1. Random Error
    2. Quantity Not Changed Since Last Entry
    3. Digits Interchanged
    4. Digits Missing

13. Quantity Used Last Month
    1. Random Error
    2. Quantity Not Changed Since Last Entry
    3. Digits Interchanged
    4. Digits Missing

14. Price Paid Last Year
    1. Random Error
    2. Quantity Not Changed Since Last Entry
    3. Digits Interchanged
    4. Digits Missing

Table 3

Tests Used for Detecting Errors in a Change-Record

Range Tests

1. Numeric Range Test
2. Alpha Range Test
3. Fixed Range Test
4. Range Test Based on Last Year's Experience
5. Range Test Based on Last Month's Experience

Consistency Tests

6. Transaction Test (Determine if the Quantity in the New Master File Equals that Quantity in the Old File Plus the Change)
7. Name Test for New Record (Does the Name Already Exist in the New Record?)
8. Number Test for New Record (Does the Number Already Exist in the Old Records?)
9. Name Test for Delete (Is There a Record With That Name in the Old File?)
10. Number Test for Delete (Is There a Record With That Number in the Old File?)
11. Balance Test (Quantity On Hand for Coordinated Parts Equal Within a Specified Tolerance.)
12. Projected Usage Test (Does the Quantity on Hand Plus Expected Delivery Minus Shrinkage Equal or Exceed the Expected Usage Next Month?)
13. Independent Varification of Data (Audit of Record Changes).

The Participant/Computer Interaction

The participant's task was to enter candidate tests for each change-record, request a cost update, review the costs, and then repeat these steps until time was up or until the participant believed the minimum-cost design had been specified. Three control keys were available, as follows:

Press F_1 to enter a design (a set of tests)

- Computer response: "enter Record [i.e., change-record] ID number"

  -- participant entered ID number

- Computer response: "enter number of tests"

  -- participant entered the number of tests

- Computer response: "enter test number 1"

  -- participant entered the first test number, from 1 to 13, and continued entering test-numbers until the entire test design had been entered.

Press F_2 to request cost of design

- Computer response: a new design-number was automatically assigned each time F_2 was pressed. Computer calculated & displayed cost of new design.

Press F_3 to view a previous design

- Computer response: "enter number of desired design"

  -- participant entered design number

- Computer response: displayed design requested.

Note that F_1, F_2, and F_3 were special function keys available on the terminal. Thus, a single key, F_1, F_2, or F_3, was pressed to command the desired function.

Calculation of Total Cost

The cost of each undetected error and the cost of using each available test were represented by elements in the cost matrices documented in Appendix C. Using the cost and error-probability matrices, the total cost of any design (set of tests) was calculated. This total cost, as noted already above, was the sum of the cost of the tests used plus the cost to the organization of the errors that went undetected in spite of the tests. This total cost may be represented as follows:

Let T_k = test k

CT_k = cost of test k

$$I_{ki} = \begin{cases} 1 & \text{if test } k \text{ is assigned to change-record } i \\ 0 & \text{if test } k \text{ is not assigned to change-record } i \end{cases}$$

The cost of all the tests assigned to all change-records is therefore given by:

$$CT = \sum_i \sum_k CT_k \times I_{ki} \qquad (1)$$

Now, to calculate the cost to the organization of all the errors that remain undetected in spite of all the tests used, let

PE_{ij} = the probability of error E_{ij}, where i is the change-record number (1-14), and j is the error-number for change-record i (note: the maximum value of j varies with i because the number of error types per inventory record is not constant. (See Table 2))

$\overline{PD}_{ij}$ = the probability of failing to detect error E_{ij}

$\bar{C}_{ij}$ = the cost to the organization if error E_{ij} occurs and is not detected.

From these definitions it follows that the expected cost $\mathcal{E}\bar{C}_{ij}$ of failing to detect error E_{ij} is just

$$\mathcal{E}\bar{C}_{ij} = PE_{ij} \times \bar{C}_{ij} \times \overline{PD}_{ij} \qquad (2)$$

The total expected cost $\mathcal{E}\bar{C}$ of all undetected errors is therefore

$$\mathcal{E}\bar{C} = \sum_i \sum_j PE_{ij} \times \bar{C}_{ij} \times \overline{PD}_{ij} \qquad (3)$$

Finally, the total cost of any design (set of tests) is just the sum of equations (1) and (3):

$$\text{Total Cost} = \sum_i \sum_j PE_{ij} \times \bar{C}_{ij} \times \overline{PD}_{ij} + \sum_i \sum_k CT_k \times I_{ki} \qquad (4)$$

Probability $\overline{PD}_{ij}$ of Failing to Detect Error E_{ij}: If there are multiple tests assigned to a change-record, an error could be detected by more than one test. Thus, the probability of error-detection by 1 or more tests must be determined. If there are N tests, there are 2^N possible detect- or fail-to-detect events. We can construct a representation of these events with the symbol T_k now used to indicate successful detection of a specific error by test k and $\overline{T}_k$ now used to indicate unsuccessful error detection. The construction is as follows:

Event: $T_1\ T_2\ T_3 \ldots T_n$; all tests detect the error

$$\left.\begin{array}{l} \text{Event: } \overline{T}_1\ T_2\ T_3 \ldots T_n \\ \text{and} \\ T_1\ \overline{T}_2\ T_3 \ldots T_n \\ T_1\ T_2\ T_3 \ldots \overline{T}_n \end{array}\right\} \text{one test fails to detect the error}$$

•

•

•

Event: $\overline{T}_1\ \overline{T}_2\ \overline{T}_3 \ldots \overline{T}_n$; all tests fail to detect the error

A method used by Boole (1854) to compute the probability of an event is to replace the logic variables with their respective probabilities. If PD_{kij} is the probability that test k will detect error j on change-record i, then $\overline{PD}_{kij} = 1-PD_{kij}$ is the probability that test k will fail to detect error j on change-record i. Applying Boole's method, the probability that no test will detect error j on change-record i becomes

$$\overline{PD}_{ij} = \overline{PD}_{1ij}\ \overline{PD}_{2ij}\ \overline{PD}_{3ij} \cdots PD_{nij}$$

or

$$\overline{PD}_{ij} = \prod_k (1-PD_{kij}).$$

After accounting for the fact that a participant can assign tests arbitrarily, this becomes

$$\overline{PD}_{ij} = \prod_k (1-PD_{kij} \times I_{ki}) \qquad (5)$$

Total Cost of a Test Design in Terms of Known Quantities: Returning to equation (4) for the total cost and substituting into it the result just obtained for $\overline{PD}_{ij}$, the total cost finally becomes

$$\text{Total Cost} = \sum_i \sum_j PE_{ij} \times \overline{C}_{ij} \times \prod_k (1-PD_{kij} \times I_{ki})$$
$$+ \sum_i \sum_k CT_k \times I_{ki} \qquad (6)$$

This total cost can also be represented as the sum of the costs of each error E_{ij}, that is:

$$\text{Total Cost} = \sum_i \sum_j \text{Total Cost}_{ij}$$

where

$$\text{Total Cost}_{ij} = PE_{ij} \times \overline{C}_{ij} \times \prod_k (1-PD_{kij} \times I_{ki})$$
$$+ \sum_k CT_k \times I_{ki}.$$

The four data matrices used in the experiment are given in Appendix C. They are:

$\overline{C}_{ij}$ = the cost to the organization of failing to detect error j on change-record i,

PE_{ij} = the probability that error j on change-record i will occur,

PD_{kij} = the probability that test k will detect error j on change-record i,

CT_k = the cost of using test k on any given change-record.

Note in Appendix C that there are 4 matrices for PD_{kij}, labeled $PD_{kij}1$ ... $PD_{kij}4$, corresponding to the pre-test and 3 experiment problems, respectively.

Corrected Total Cost: Differential, or Corrected Total Cost (CTC), was used in the data analysis. The CTC was equal to the total cost given above minus the minimum cost possible, obtained when the optimal set of tests was specified.

Design of the Experiment

The experiment used a repeated-measures Latin Square design (Plan #9 cited in Winer, 1971, pp. 727-736). The design is shown in Table 4. The factors investigated were:

a. 3 levels of problem-complexity,* where each level required a different amount of effort to correctly specify the problem-solution.

b. 3 levels of costing-aids*, where each level required that a different amount of information be provided by the user to correctly specify the solution.

Referring now to Table 5, the numbers in column one are the numbers assigned to the participants. The numbers in columns two, three, and four refer to the problem-number and aid-number, respectively. The numbers in column five refer to the groups to which the participants were assigned and, in column six, to the order of problem presentation.

Each participant was given:

1. A description, via video tape, of the experiment-task along with an illustrative solution.

2. A description, via video tape, of the procedure for entering data into the computer.

3. A pre-test problem -- a low-complexity problem with the Level 2 costing-aid.

4. Test Problem #X, Aid Level 1.

* Measures of problem-complexity and costing-aid level are discussed in subsequent sections of this report.

Table 4

Experiment Design
Repeated-Measures Latin Square

	A_1B_j	A_2B_j	A_3B_j
G_1	A_1B_2	A_2B_3	A_3B_1
G_2	A_1B_1	A_2B_2	A_3B_3
G_3	A_1B_3	A_2B_1	A_3B_2

A_i is a level of problem-complexity.

B_j is a level of costing-aid.

Each combination of A_i and B_j is an experiment cell.

G_k is a group of 12 participants.

Table 5

Experiment Plan

	Problem # - Aid #				
Participant #	Session 2	Session 3	Session 4	Group	Order
1	2-3	1-2	3-1	1	5
2	2-2	1-1	3-3	2	5
3	2-1	1-3	3-2	3	5
4	1-2	3-1	2-3	1	4
5	1-1	3-3	2-2	2	4
6	1-3	3-2	2-1	3	4
7	1-2	2-3	3-1	1	1
8	1-1	2-2	3-3	2	1
9	1-3	2-1	3-2	3	1
10	3-1	1-2	2-3	1	6
11	3-3	1-1	2-2	2	6
12	3-2	1-3	2-1	3	6
13	3-1	2-3	1-2	1	3
14	3-3	2-2	1-1	2	3
15	3-2	2-1	1-3	3	3
16	2-3	3-1	1-2	1	2
17	2-2	3-3	1-1	2	2
18	2-1	3-2	1-3	3	2
19	1-2	2-3	3-1	1	1
20	1-1	2-2	3-3	2	1
21	1-3	2-1	3-2	3	1
22	2-3	3-1	1-2	1	2
23	2-2	3-3	1-1	2	2
24	2-1	3-2	1-3	3	2
25	1-2	3-1	2-3	1	4
26	1-1	3-3	2-2	2	4
27	1-3	3-2	2-1	3	4
28	3-1	1-2	2-3	1	6
29	3-3	1-1	2-2	2	6
30	3-2	1-3	2-1	3	6
31	3-1	2-3	1-2	1	3
32	3-3	2-2	1-1	2	3
33	3-2	2-1	1-3	3	3
34	2-3	1-2	3-1	1	5
35	2-2	1-1	3-3	2	5
36	2-1	1-3	3-2	3	5

5. Test Problem #Y, Aid Level m.

6. Test Problem #Z, Aid Level n.

Note that values for X, Y, Z, l, m, n were taken from Table 5 in accordance with the participant's group number.

Experiment Problems: A pre-test and three experiment problems were used. Problem-complexity was controlled by the number of tests available for assignment to each change-record. In the pre-test, tests 1 through 4 were available. In experiment problems 1, 2, and 3, tests 1 through 3, 1 through 8, and 1 through 13, respectively, were available. Tables 6 through 9 give for each problem the tests permitted, the number of possible test combinations*, the total cost when no tests were specified, the total cost when the optimum tests were specified, and the optimum tests and the total costs for each individual change-record. In all cases, a maximum of 6 tests at a time could be specified for a given change-record. This limitation, which was imposed to avoid overcrowding of the display, had little real impact because in all problems specification of more than three tests resulted in high costs.

Costing-Aids: Three levels of costing-aids were used in the experiment. The Level 1 costing-aid consisted of only the total cost of the present design, i.e., of the present set of specified tests. With the Level 1 costing-aid, only the total cost over all change-records, and not for individual change-records, was given. The terminal display presented to the participant for this costing-aid is given in Table 10.

The Level 2 costing-aid consisted of the total cost of the present design plus the total cost of each component of the design, i.e., of the tests specified for each change-record. Table 11 shows the terminal display for this aid.

The Level 3 costing-aid provided the same information as the Level 2 aid and, in addition, provided a listing of previous designs, ordered according to cost, for each change-record. Table 12 shows the terminal display for this aid.

* The number of possible test combinations was given by a sum of binomial coefficients. Thus, if there were M tests available and a maximum of N tests at a time could be chosen, then the total number of possible test combinations was $\sum_{n=0}^{N} \binom{M}{n}$, where $\binom{M}{n} = \frac{M!}{n!\,M-n!}$

Table 6

Pre-Test Problem

1.	Tests permitted	1, 2, 3, 4
2.	Number of possible test combinations	16 *
3.	Total cost when no tests were specified	$18,465.00
4.	Total cost when optimum tests were specified for each change-record	$ 6,960.23

Change-Record Number	Optimum Set of Tests	Change-Record Total Cost for Optimum Tests
1	4	367.38
2	2,4	545.24
3	3	1,397.50
4	4	505.00
5	3,4	385.55
6	3	264.10
7	3	184.75
8	0	97.50
9	1	349.70
10	3,4	453.81
11	1	1,102.25
12	3	527.95
13	3	190.15
14	1,3	589.35

* Given by $\sum_{n=0}^{4} \binom{4}{n}$, a sum of binomial coefficients

Table 7

Experiment Problem-Complexity Level 1

1.	Tests permitted	1, 2, 3
2.	Number of possible test combinations	8*
3.	Total cost when no tests were specified	$18,465.00
4.	Total cost when optimum tests were specified for each change-record	$ 9,300.28

Change-Record Number	Optimum Set of Tests	Change-Record Total Cost for Optimum Tests
1	3	771.85
2	2,3	428.84
3	1	1595.00
4	1	1100.00
5	3	445.00
6	3	418.38
7	1	342.50
8	0	97.50
9	3	456.25
10	1,3	682.03
11	1,3	769.22
12	1	630.50
13	1,3	390.06
14	1,3	1173.15

* Given by $\sum_{n=0}^{3} \binom{3}{n}$, a sum of binomial coefficients

Table 8

Experiment Problem—Complexity Level 2

1.	Tests Permitted	1 through 8 (no more than 6 at a time)
2.	Number of possible test combinations	247*
3.	Total cost when no tests were specified	$18,465.00
4.	Total cost when optimum tests were specified for each change-record	$6,503.62

Change-Record Number	Optimum Set of Tests	Change-Record Total Cost for Optimum Tests
1	8	211.00
2	5	387.25
3	3,4	1,377.87
4	3,7	381.50
5	3	175.00
6	3,8	256.75
7	3,8	372.50
8	0	97.50
9	1	263.30
10	4,7	429.46
11	3,7	62.28
12	3	395.65
13	4	337.15
14	4,5	800.85

* Given by $\sum_{n=0}^{6}\binom{8}{n}$, a sum of binomial coefficients, limited to 6 tests.

Table 9

Experiment Problem-Complexity Level 3

1. Tests Permitted — 1 through 13 (no more than 6 at a time)

2. Number of possible test combinations — 4668*

3. Total costs when no tests were specified — $18,465.00

4. Total cost when optimum tests were specified for each change-record — $7,855.49

Change-Record Number	Optimum Set of Tests	Change-Record Total Cost for Optimum Tests
1	8	331.30
2	5,7	453.58
3	4,11	1,207.50
4	3	1,135.00
5	3,7	365.00
6	3,7	359.96
7	3	520.00
8	0	97.50
9	3,5	428.59
10	3,4	625.06
11	7,8,9	352.28
12	1,3	772.07
13	7,8,9	352.28
14	3,4	855.37

* Given by $\sum_{n=0}^{6} \binom{13}{n}$, a sum of binomial coefficients, limited to 6 tests.

Table 10

Costing-Aid Level 1 Display

	Change-Record Field	Tests
1.	Part Number	4 0 0 0 0 0
2.	Part Name	2 4 0 0 0 0
3.	Quantity on Hand	3 0 0 0 0 0
4.	Quantity on Order	4 0 0 0 0 0
5.	Percent Damaged in Last Order	3 4 0 0 0 0
6.	Quantity Expected in 1 Month	3 0 0 0 0 0
7.	Delivery Time When Ordering Now	3 0 0 0 0 0
8.	Quantity Discount	0 0 0 0 0 0
9.	Unit Price	1 0 0 0 0 0
10.	Quantity Used This Year	3 4 0 0 0 0
11.	Quantity Used This Month	1 0 0 0 0 0
12.	Quantity Used Last Year	3 0 0 0 0 0
13.	Quantity Used Last Month	3 0 0 0 0 0
14.	Price Paid Last Year	1 3 0 0 0 0

Total Cost is $6,960.23

Table 11

Costing-Aid Level 2 Display

	Change-Record Field	Record Cost	Tests
1.	Part Number	367.38	4 0 0 0 0 0
2.	Part Name	545.24	2 4 0 0 0 0
3.	Quantity on Hand	1397.50	3 0 0 0 0 0
4.	Quantity on Order	505.00	4 0 0 0 0 0
5.	Percent Damaged in Last Order	385.55	3 4 0 0 0 0
6.	Quantity Expected in 1 Month	264.10	3 0 0 0 0 0
7.	Delivery Time When Ordering Now	184.75	3 0 0 0 0 0
8.	Quantity Discount	97.50	0 0 0 0 0 0
9.	Unit Price	349.70	1 0 0 0 0 0
10.	Quantity Used This Year	453.81	3 4 0 0 0 0
11.	Quantity Used This Month	1102.25	1 0 0 0 0 0
12.	Quantity Used Last Year	527.95	3 0 0 0 0 0
13.	Quantity Used Last Month	190.15	3 0 0 0 0 0
14.	Price Paid Last Year	589.35	1 3 0 0 0 0

Total Cost is $ 6,960.23

Table 12

Costing-Aid Level 3 Display

	Change-Record Field	Record Cost	Tests
1.	Part Number	367.38	4 0 0 0 0 0
2.	Part Name	545.24	2 4 0 0 0 0
3.	Quantity on Hand	1397.50	3 0 0 0 0 0
4.	Quantity on Order	505.00	4 0 0 0 0 0
5.	Percent Damaged in Last Order	385.55	3 4 0 0 0 0
6.	Quantity Expected in 1 Month	264.10	3 0 0 0 0 0
7.	Delivery Time When Ordering Now	184.75	3 0 0 0 0 0
8.	Quantity Discount	97.50	0 0 0 0 0 0
9.	Unit Price	349.70	1 0 0 0 0 0
10.	Quantity Used This Year	453.81	3 4 0 0 0 0
11.	Quantity Used This Month	1102.25	1 0 0 0 0 0
12.	Quantity Used Last Year	527.95	3 0 0 0 0 0
13.	Quantity Used Last Month	190.15	3 0 0 0 0 0
14.	Price Paid Last Year	589.35	1 3 0 0 0 0

Total Cost is $ 6,960.23

Computer Prompt: "Enter Record ID Number for Extended Analysis"

Participant enters number: 4

Computer Responds: "Least Cost Designs for Record 4 are:

Design Number 37 Cost is 505.00

Design Number 13 Cost is 819.00

Design Number 8 Cost is 1102.23

Design Number 41 Cost is 1303.15

Remember the best Design will consist of only the best Records".

Measures

Performance Measures

For each of the four experiment problems there was an optimal set of tests for each change-record — optimal in the sense that specification of those tests resulted in the least total cost. One performance measure, termed the "cost measure", was the actual cost achieved. Another measure was the total cost achieved minus the least total cost possible for that problem, i.e., what was defined earlier, in the section on "Calculation of Total Cost", as the Corrected Total Cost, or CTC. The least total cost for each problem is given in Tables 6 through 9.

Strategy Measures

The following measures were developed in an attempt to capture the strategy used by the participants.

1. Number of Designs

   Equal to the total number of designs (sets of tests) specified for the problem. Each design could involve specification of multiple tests for all change-records. Each new design could also involve modifications of the previous tests for one or more change-records.

2. Time

   Equal to the time, in minutes, used for the problem. Maximum time permitted was 60 minutes.

3. Change-Record Selection

   The purpose of this measure was to determine whether the participant tended to work on the most-costly change-record when specifying a set of tests. Before each set of tests was analyzed, the total cost of each change-record was determined, and the results were ordered with the most-costly first. Then, for each new set of tests specified for change-record X, for example, a partial score was developed according

to the ordering of change-record X. If change-record X had the greatest cost prior to specification of the new set of tests (i.e., order #1), then the partial score was 1. If the change-record ordering was "10", the partial score was a 10, etc.

These partial scores were summed as each new set of tests was specified. A low value of the sum indicated that the participant tended to concentrate his or her work on the most-costly change-record, where as a high score indicated that the participant tended to work on change-records with less-than-the-greatest cost. This strategy measure was termed "Change-Record Selection" (CRS).

4. Average Change-Record Selection

The sum value, described above, developed as the Change-Record Selection strategy-measure was normalized by dividing by the total number of test-set modifications entered by the participant. This normalized strategy-measure was termed the "Average Change-Record Selection" (ACRS).

5. Number of Designs Viewed

The number of designs viewed was the total number of times a previous design was requested.

6. Average Number of Test-Sets per Change-Record

The number of test-sets per change-record specified by the participant was determined. Then the average of those values over the 14 change-records was computed and used for a strategy-measure.

7. Total Number of Test-Sets

The total number of test-sets specified by the participant for all change-records was another strategy-measure.

8. Consistency

This measure was developed to ascertain the tendency of the participant to work on one change-record at a time, i.e., by specifying and evaluating various test-sets for one change-record before preceeding to the next change-record. This strategy-measure, termed "Consistency", developed a partial score by giving 1 point when a test-set was first specified for a change-record, 2 points for the second test-set (if different from the first) for that same change-record, 4 points for the third test-set (if different from the first and second) for that same change-record, 8 points for the fourth test-set ... etc. When a test-set was specified for a change-record different from the previous change-record, the partial score was reset to 1.

This consistency measure had a large value when the participant tended to specify, and presumably to evaluate, multiple test-sets for a given change-record. Conversely, if the participant tended to skip from one change-record to another, the measure value was low.

Participants

The participants were individuals experienced in some type of inventory-control problem but were not experienced software analysts or programmers. All participants were obtained via the newspaper ad reproduced in Appendix D. The participant demographics were as follows:

1. All participants were high school graduates.

2. There were 10 female participants, with ages ranging from 25 years to 47 years.

3. The average female participant's age was 33 years.

4. There were 26 male participants, with ages ranging from 22 years to 62 years.

5. The average male participant's age was 33 years.

6. The range of years of higher education was 0 years to 8 years.

7. The average number of years of higher education was 5 years.

8. The educational majors included: Management (9), Administration (5), Business (5), Marketing (3), and Engineering (2).

9. The work experience of the participants included: department stores (6); small businesses (6); consulting (5); sales, management and production analysts (4); Armed Forces (4); sales (3); fast food chains (3); and supervision (3).

Procedure

Participants were scheduled for either a morning session, beginning at 8:00 a.m., or an afternoon session, beginning at 1:00 p.m. When a participant arrived, he/she was asked to fill out a biographical questionnaire (the questionnaire is given in Appendix E) to verify that the participant's experience satisfied the experiment entrance criteria and to obtain additional information regarding the level of experience in his/her particular field. If the participant's experience did not satisfy the criteria, he/she was not used in the experiment. If the participant's experience satified the experiment entrance criteria, the experiment was briefly explained, and the participant was provided with a consent form (Appendix E), having been assured that no personal risk was involved. The participant then signed this form to indicate that he/she understood these arrangements.

The participant was next seated in the experiment room. The room was approximately 12 x 16 feet in size, with a video tape recorder and video monitor located on one table, and a computer and terminal on a separate table. Participants were asked to make themselves comfortable and to adjust the light and ventilation to their satisfaction.

Instructions for the experiment were presented in two parts, both of which were on video tape. The first part described the experiment problem and gave a method for solving the problem, including an example-solution. The second part described how to enter data into the computer and also included an illustrative problem-solution. Since this second portion of the instructions employed a dynamic display of the operation of the computer, it cannot be presented here.

After the instructions were presented, the participant was seated in front of the computer terminal. He/she was asked to use the numbered keys labeled 0-9 and the RETURN key, as well as three special-function keys. In addition, a pad of paper and pencils were provided for taking notes. These sheets were kept in each participant's file for reference.

Participants were told that up to one hour was allotted for each problem, and that the computer would automatically stop the problem when the time limit was reached. Participants were permitted to take a short break between test problems if they desired.

Data Collection

Problem-solutions were entered into the computer by the participants. The computer recorded all keystrokes, as well as the time of execution of each. This method of collecting data allowed a printout to be generated listing detailed information about each participant's activities during the experiment.

RESULTS

Mean Score

The mean scores for the nine experiment cells are presented in Figure 3. The scores are Corrected Total Cost (CTC), i.e., the total cost minus the least possible cost for the problem. These data show a direct increase in CTC with an increase in problem-complexity. Further, the data show an increase in CTC with increasing costing-aid level for problem-complexity 1 & 2, but show a reverse trend for problem level 3, the most complex problem.

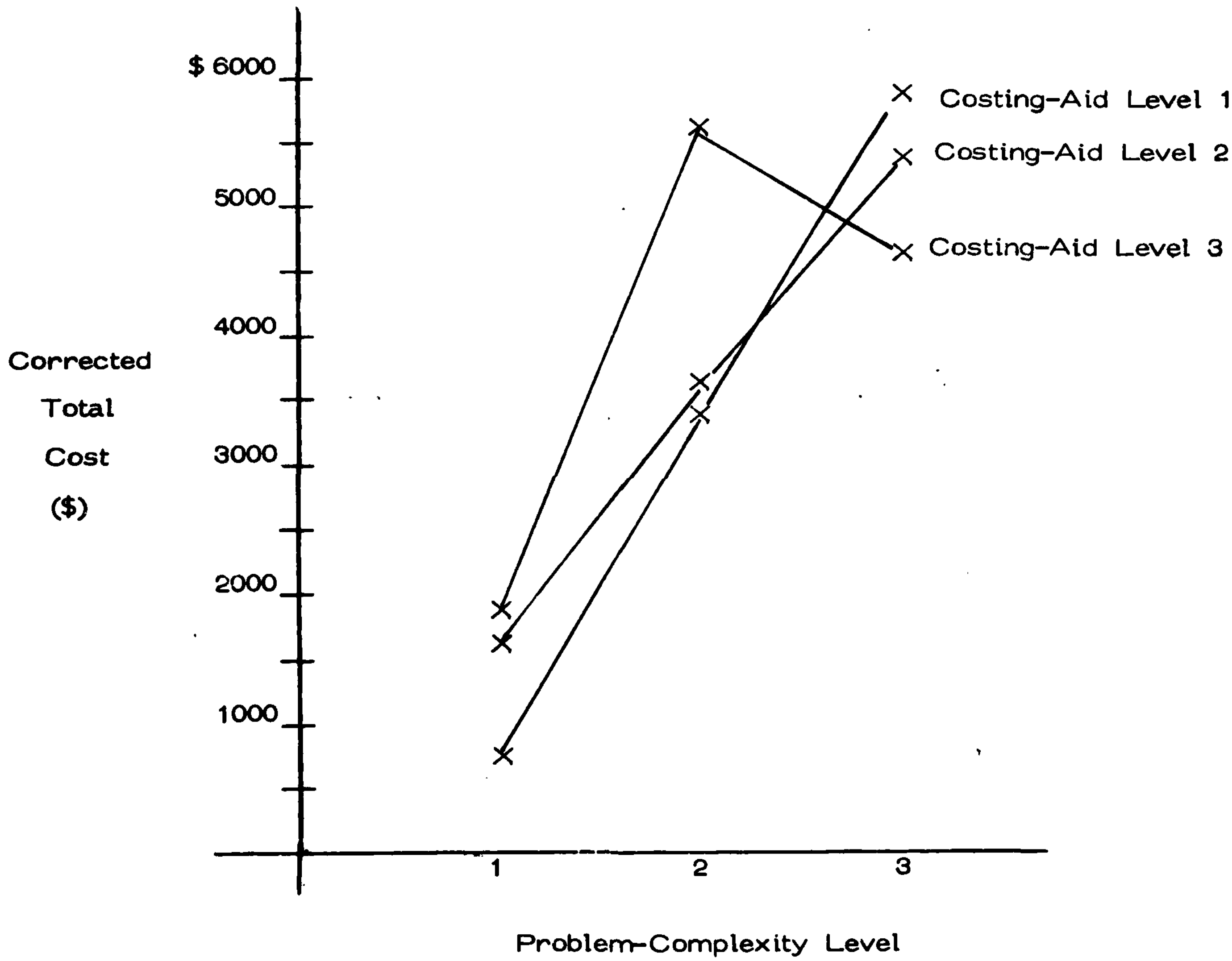


Figure 3. Total Cost vs. Costing-Aids and Problem-Complexity

Analysis #1: Test of Variance Homog ety

Score variances over the three levels of problem-complexity and the three levels of costing-aids are given in Table 13. Since the variances differed, especially as problem-complexity was varied, tests of variance homogeniety were used to test the hypothesis of equal variances. The Burr-Foster Q-test and Bartlett's test (Anderson, McLean 1974) were used.

Results. The Burr-Foster Q-test uses the statistic

$$q = (S_1^4 + \ldots + S_p^4) \;/\; (S_1^2 + \ldots + S_p^2)^2$$

where S_i^2 is the level i variance and p is the number of levels for the problem at hand. In this experiment, p = 3.
The statistic for the three levels of problem-complexity was

$$q = .3786,$$

and for the three levels of costing-aids was

$$q = .3363.$$

Critical values of q from the Q table (Anderson, McLean 1974) are the following.

crit (P=3, df=20, α = .01) = .512

crit (P=3, df=20, α = .001) = .596

crit (P=3, df=60, α = .01) = .367

crit (P=3, df=60, α = .001)= .384

Note that these critical values depend on the degrees of freedom. In this experiment df=35 (=36-1). Based on a straight-line interpolation:

crit (P=3, df=35, α = .01) = .397

Since q for both treatments was less than .397, there is no reason to believe that the variance was not homogeneous.

Table 13

Experimental Factor Variances

Variances: Three Problem-Complexity Levels

LEVEL	VARIANCE
1	2,737,596
2	5,664,256
3	7,466,647

Variances: Three Costing-Aids

LEVEL	VARIANCE
1	7,560,491
2	7,202,013
3	8,940,535

Since interpolation over an extended range, as above, can be risky, a Bartlett test was also used to test the equal-variance hypothesis.

The Bartlett test (Anderson, et al.) uses the statistic:

$$\lambda^2 = M/C$$

where

$$M = 2.3026\,(df)\,(K \log \overline{S^2} - \Sigma \log S^2)$$

$$C = 1 + \frac{K+1}{3\ df\ K}$$

and where

df = the degrees of freedom per variance

K = number of levels

S^2 = variance at each level

$\overline{S^2}$ = average variance over the K levels

For the treatment with greatest range of variance (the three levels of problem-complexity)

$K = 3$

$df = 2$ (= the number of levels of problem-complexity minus 1)

$\log \overline{S^2} = 6.723$

$\Sigma \log S^2 = 20.063$

and, thus,

$\lambda^2 = 8.531$

Critical values from the λ^2 tables for df = 2 are:

$\lambda^2\ (df{=}2,\ \alpha \leq .001) = 13.82$

$\lambda^2\ (df{=}2,\ \alpha \leq .01) = 9.21$

As with the Burr-Foster Q test, there was no reason to reject the hypothesis that the variances were equal.

Analysis #2: Analysis of Variance

Purpose. To determine whether problem-complexity, the costing-aids, or participant group significantly affected CTC.

Method. Analysis of variance.

Results. Table 14 presents the results of the analysis of variance. These results indicate that problem-complexity was highly significant and that the group factor was also significant. The costing-aids and the interaction effects were not significant.

Analysis #3: Student-Newman-Keuls Test

Purpose. To determine whether the experiment factor levels and/or the experiment cells produced statistically different results.

Method. A Student-Newman-Keuls (SNK) (Sokal, Rohlf 1969) posterior comparison-of-means test was used to test for significance of the effects of the factor levels and experiment cells.

Results. Table 15 presents the results of the SNK test applied to the three levels of the two experiment factors. The results show that the effect of the first problem-complexity level was significantly different from that of the second and third levels, but that the level-two and level-three effects were not significantly different. These results also show that the effects of the costing-aid levels were not significantly different.

Table 16 provides the results of the SNK test applied to the nine experiment cells. These results show that the first level of problem-complexity in combination with any level of costing-aid had effects that were significantly different from those of the third problem-complexity level combined with the third level of costing-aid, as well as from the second problem-complexity level combined with the third level of costing-aid.

Table 17 provides the results the SNK test applied to the three participant groups. The results do not provide any evidence that corresponding cells differed significantly according to group.

Table 14

Analysis of Variance

Source of Variation	Sum Sq	DF	Mean Sq	F-Ratio
Problem-Complexity	289.63	2	144.81	27.28***
Costing-Aids	7.57	2	3.78	.71*
Group	29.93	2	14.96	2.82
Residue	7.85	2	3.92	.74
Within Cell	525.50	99	5.30	
Total	860.48			

*** $P \leq .001$
* $P < .10$

Table 15

SNK Test Applied to Levels of Experiment Factors

		Problem-Complexity Level					
		Level 1	Level 2	Level 3	MEAN	1st Difference	2nd Difference
Costing-Aid Level	Level 1	.790	3.375	5.740	3.30		
						0.12^{NS}	
	Level 2	1.529	3.550	5.210	3.42		0.66^{NS}
						0.48^{NS}	
	Level 3	1.629	5.490	4.620	3.90		

Mean	1.31	4.18	5.19
1st difference	2.88**	1.01^{NS}	
2nd difference	3.89^{**}		

Least Significant Range (LSR) for 2 Means = 2.258

LSR for 3 Means = 2.563

** for P $\leq$ .01

Table 16

SNK Test Applied to Experiment Cells

Problem-Complexity Level	Costing-Aid Level	Mean				
1	1	.790	.790			
1	2	1.529	NS	1.529		
1	3	1.629	NS	NS	1.629	
2	1	3.375	NS	NS	NS	3.375
2	2	3.550	NS	NS	NS	NS
3	3	4.620	*	*	*	NS
3	2	5.210	*	*	*	NS
2	3	5.490	*	*	*	NS
3	1	5.740	*	*	*	NS

For $P \leq .01$, the least significant range LSR for 9 Means = 3.09,

LSR for 8 Means = 3.04
LSR for 7 Means = 2.97
LSR for 6 Means = 2.89
LSR for 5 Means = 2.80

* Significantly different from cell at top of column for $P \leq .01$ level.

Table 17

SNK Applied to Participant Groups

Group	Mean	1st Difference	2nd Difference
1	4.253		
		$.85^{NS}$	
3	3.402		1.28^{NS}
		$.43^{NS}$	
2	2.968		

For $P \leq .01$, the least significant range (LSR) for 2 means = 2.258, LSR for 3 Means = 2.563

Analysis #4: Correlation/Regression Analyses of Experiment Factors

Purpose. To determine the correlations among the experiment variables and the ability of the experiment factors to predict CTC.

Method. Correlation analysis among the experiment variables, and with the CTC score. In addition, univariate and step-wise linear multivariate regressions with the experiment variables.

Results. Correlations among the experiment (independent) variables are given, along with strategy factors, in Table 22 (Analyses 6). Tables 18 and 19 present the univariate and multivariate regressions using CTC as the dependent variable. The univariate regression showed that only pre-test score and problem-complexity were statistically significant, explaining 21% and 30% of the variance, respectively.

Likewise, the multivariate regression analysis revealed that pre-test score, problem-complexity and the number of designs specified by the participant were significant. In neither the univariate nor the multivariate analyses were the costing-aids found to be significant.

Analysis #5: Correlation/Regression Analyses of Demographic Factors

Purpose. To determine the correlations among the demographic factors and the ability of the demographic factors to predict CTC.

Method. Correlation analysis among the demographic factors and between the demographic factors and CTC. In addition, univariate and step-wise linear regression analyses with the demographic factors as the independent variables and CTC as the dependent variable.

Results. Table 20 gives the results of the correlation analysis among the demographic factors. The correlations are moderate, thus allowing any combination of factors as independent variables in multivariate regressions. Table 21 presents the results of the univariate regression analysis. Only two factors, "years-of-higher-education" and "an Economics Degree", were found to be significant, explaining 18% and 11% of the CTC variance, respectively.

Table 18

Correlation and Univariate Regression

Effects of Experiment Factors

Dependent Variable: CTC

Independent Variable	Correlation With CTC	Regression			
		Coefficient	Variance Explained	t-Value	P ≤
Score on Pre-test	.46	.61	21.9%	5.4	.001
Session Number	-.05	-1.98	.3%	- .5	NS
Problem Complexity	.55	19.38	30.7%	6.8	.001
Costing-Aid	.08	3.06	.8%	.90	NS

Table 19

Multivariate Regression

Effects of Experiment Factors
Dependent Variable: CTC

Independent Variable	Regression Coefficient	Regression Variance Explained	t-Value	P ≤
Score on Pre-test	23.75	57.2%	9.15	.001
Problem-Complexity	.61		7.29	.001

Analysis of Variance Table

Source	Sum Sq	DF	Mean Sq	F-Ratio	P ≤
Regression	50514.4	3	16838.1	46.32	.001
Residue	37798.4	104	363.4		

Table 20

Correlation Among Demographic Factors

	1	2	3	4	5	6	7	8	9	10	11	12
1. Age		.00	-.34	.19	.05	.17	-.11	.11	-.07	.11	.11	.55
2. Years of Education			.26	-.10	.07	.54	.01	-.06	-.14	-.04	-.10	.03
3. Business Degree				-.40	-.23	-.07	-.07	-.43	.13	.14	-.12	-.03
4. Engineering Degree					-.05	-.07	-.07	-.16	.12	-.01	-.14	.02
5. Education Degree						-.04	-.04	-.09	-.15	-.11	.29	.13
6. Economics Degree							-.06	-.13	-.22	-.16	.02	.05
7. Psychology Degree								-.13	-.12	-.05	-.09	-.09
8. Other Degree									.15	.24	.09	-.09
9. No. of Prog. Language										.74	.22	.10
10. No. of Written Programs											-.09	.00
11. No. of Mo. in Data Processing												.25
12. No. of Yrs. in Computer Science												

Table 21

Correlation Analysis and Univariate Regression

Effect of Demographic Factors on CTC

Independent Variable	Correlation With CTC	Regression			P≤
		Coefficient	Variance Explained	t-Value	
Subject Age	.150	.363	2.2%	.884	NS
Years of Higher Education	.431	4.272	18.6%	2.788	.01
Business Degree	.238	10.867	5.7%	1.429	NS
Engineering Degree	-.188	-14.936	3.5%	-1.118	NS
Education Degree	-.206	-27.522	4.3%	-1.229	NS
Economics Degree	.343	32.814	11.8%	2.128	.05
Psychology Degree	.190	18.202	3.6%	1.130	NS
Other Degree	.078	4.087	.6%	.453	NS
No. of Programming Lang. Known	-.232	- 3.996	5.4%	-1.391	NS
No. of Programs Written	-.028	- .036	.1%	- .162	NS
Months Exper. in Data Processing	-.170	- .062	2.9%	-1.007	NS
Years Exper. in Computer Science	.074	.312	.6%	.434	NS

Analysis #6: Correlation/Regression Analyses of Strategy Measures and Experiment Factors

Purpose. To determine the correlations among the strategy measures and the experiment factors, and the ability of the strategy measures and the experiment factors to predict CTC.

Method. Correlation analysis among the strategy measures and the experiment factors, and also between those measures and factors and CTC. In addition, univariate and step-wise linear regression analyses with the strategy measures and experiment factors as independent variables and CTC as the dependent variable.

Result. Table 22 gives the results of the correlation analysis. The factors "number of designs", "average number of test-sets per change-record" and "total number of test-sets" were highly correlated and therefore could not be used together as independent variables in multivariate regressions. Tables 23 and 24 present the results of the univariate and multivariate regression analyses, respectively. In the univariate analysis, "problem-complexity", "pre-test score", "time", "average change-record selection" and "consistency" were significant. In the multivariate regressions, the factors noted above plus "change-record selection" were found to be significant predictors of CTC.

Table 22

Correlations Among Strategy & Experiment Factors

	1	2	3	4	5	6	7	8	9	10	11	12
1. Session		.00	.00	.09	-.04	.10	-.04	-.04	.13	.13	.05	.00
2. Problem-Complexity			.00	-.08	.29	-.12	-.04	-.11	.03	.03	.15	.00
3. Costing-Aid				-.26	.02	-.24	-.06	-.14	-.23	-.23	-.01	.00
4. Number of Designs					.04	.85	-.31	-.16	.91	.91	.34	-.19
5. Time						-.00	-.02	.14	.00	.00	.06	.18
6. Change-Record Selection							.43	-.15	.82	.82	.40	-.03
7. Average Change-Record Selection								-.18	.21	.21	.17	.01
8. No. of Designs Viewed									-.17	-.17	-.13	.14
9. Ave. Number of Test Sets Per Record										1.00	.51	-.03
10. Total Number of Test Sets											.51	-.03
11. Consistency												.10
12. Pre-test Score												

Table 23

Correlation Analysis and Univariate Regression
Effects of Experiment and Strategy Factors on CTC

Independent Variable	Correlation With CTC	Regression Coefficient	Regression Variance Explained	Regression t-Value	P≤
1. Session No.	-.05	-1.981	.3%	- .58	NS
2. Problem Complexity	.55	19.389	30.7%	6.84	.001
3. Cost	.08	3.069	.8%	.90	NS
4. No. of Designs	-.17	- .142	3.0%	-1.79	NS
5. Time	.25	.817	6.6%	2.73	.01
6. Change-Record Selection	-.09	- .766	.8%	-0.95	NS
7. Ave. Change-Record Selection	.15	2.51	2.5%	1.66	NS
8. No. of Designs Viewed	.08	.183	.6%	.83	NS
9. Ave. No. of Test-Sets Per Change-Record	.02	.345	.1%	.06	NS
10. Total No. of Test-Sets	.02	.025	.1%	.024	NS
11. Consistency	.29	.021	8.9%	3.22	.001
12. Pre-Test Score	.46	.610	21.9%	5.45	.001

Table 24

Multivariate Regression
Effects of Experiment and Strategy Factors on CTC

Independent Variable	Coefficient	Variance Explained	t-Value	P ≤
Problem	18.45		7.95	.001
Pre-Test Score	.588	55.4%	6.85	.001
Consistency	.012		2.59	.02

Analysis of Variance Table

Source	Sum Sq	DF	Mean Sq	F-Ratio	P≤
Regression	48940.29	3	16313.43	43.09	.001
Residue	39372.56	104	378.58		

DISCUSSION

Effect of Costing-Aids on Performance

The costing-aids used in the experiment did not result in a statistically significant effect on a participant's ability to identify the least-cost system. But these costing-aids were selected only as models of the information typically available when a non-programming user attempts to develop RS while working with a software expert. Costing-aid Level 1 corresponded to the case in which only the total cost of a software product is made available even though there are identifiable parts to the product whose specifications are manipulated by users in an attempt to identify the least-cost RS. When the cost of individual parts of the product are not available, the user cannot direct his or her efforts to the most-costly parts. Such a user could, with bad luck, work on a part that has little cost-impact and thus neglect more fruitful areas.

Costing-aid Level 2 provided both the product's total cost and the total cost of each component. With this additional information, the participant could work on the most-costly items first before proceeding to the less-costly items. Although this strategy did not guarantee that the result would be the least-cost product, it was thought that it would help to reduce the total cost rapidly. This is what participants were instructed to do, but often neglected.

Costing-aid Level 3 provided the total cost as well as the component-part total costs, as did Level 2; in addition, it provided a listing of previous designs, ordered according to cost, for each component part, i.e., for each change-record. This additional information presented the best design, and the tests used to achieve that design, for each change-record of interest to the participant. This is analogous to the information available in a non-computer system when careful, systematic records are kept with paper and pencil of each trial design. Such a non-computer system could involve a user talking with a software expert, where the user presents trial designs and the expert computes the cost of each.

Although there was a tendency for costing-aid Levels 2 & 3 to degrade performance at problem Levels 1 & 2, and to improve performance at problem Level 3, the statistical analyses demonstrated that the costing-aids did not support superior, or even satisfactory

performance. Nevertheless, it clearly is important for a user to discover which features are available at low cost and to include them in the final RS, as well as to discover those features that are high cost and to exclude them. Consequently, one must consider alternative systems. Some of these are:

1. Train software experts also to be expert in an application area, namely, the user's problem area. This alternative is often used now by default - in cases where the user cannot develop RS or provides incomplete RS -- but is obviously the least desirable alternative because the user lacks control over the product.

2. Develop a method for the user to specify the trade-off criteria he or she employs in evaluating alternative RS and in selecting the final RS. With these criteria specified, the RS could then be constructed by the software designer. This method involves a totally different concept from that investigated here or presently used. It will not be investigated in this project until the third alternative, below, has been investigated.

3. Develop a method by which the user can develop more complete and more cost-effective RS. The results reported here, where the ability of the managers to develop cost-effective RS was found to be poor, suggest that the new aids must not only provide automatic calculations of the least-cost system, but must also guide the user in collecting all the necessary data.

Alternative #3 is being pursued currently with the design of improved user-aids.

Effect of Problem-Complexity

The ANOVA, SNK, and regression analyses showed that problem-complexity produced significantly different experiment results. The results of the SNK analysis suggest that there was a significant difference in effect between problem Levels 1 and 2, and 1 and 3, but not between Levels 2 and 3. Future studies will involve a greater difference in problem-complexity between Levels 2 and 3.

Effect of Demographic Factors

The correlation and univariate regression analyses to establish the correlations between the demographic factors and CTC as well as the factors' ability to predict CTC, revealed that only two factors, "years-of-higher education" and "an Economics degree", resulted in statistically significant results. These factors accounted for 18% and 12% of the variance, respectively. One surprising result was that both factors were positively correlated with CTC, i.e., an increase in years-of-higher-education or the taking of an Economics degree was associated with an increase in the system cost! The reason for this association is not known at present. Further studies are planned to determine whether there are related demographic factors that would explain these results.

CONCLUSIONS AND PLANS

The results suggest that the user/software interchange without the benefit of sophisticated costing-aids does not result in RS capable of guaranteeing a least-cost system. It is possible that the use of an actual software expert instead of a simulated costing expert might improve performance, i.e., that improvement might have to be directed by the software expert rather left to the user. Since the goal here was to develop and test aids to help a user working with a software expert in developing quality RS, we conclude that sophisticated aids must be developed that will assist in insuring that all necessary information is obtained that will provide the necessary calculations automatically.

APPENDIX A

Available Tools/Models/Design Methods

Richhart (1983) identified tools, models, and design methods for RS and software development. Those devices are listed in this Appendix.

Available Tools/Models/Systems

Non-Automated Tools

HIPO -	Hierarchical Input, Process & Output functional design diagrams for modeling programming projects in terms of levels of systems, programs, and modules. Used to represent a system as a hierarchy of input/process/output modules.
DFD -	Data Flow Diagrams These indicate how data flows and is transformed from one kind of data item into another as it passes through data-analyzing systems.
Nassi-Schneiderman Diagrams	A low-level design technique for graphical description of the structure and statements in a class of well-structured programs. (Nassi, Schneiderman 1973).
Structure - Diagrams/ Charts	A hierarchical chart showing function-modules and the functional flow and calling sequence between modules. Some users also include the functional inputs and outputs for the modules being called.
Warnier- - Orr Diagrams	Show an expansion of the system on the left side of the page into increasingly greater detail toward the right side of the page. Warnier's techniques have been expanded and marketed in the U.S. by Ken Orr and Associates. (Orr, 1981) (King, 1981).

Semiautomated Tools

Each of these involves some form of graphical notation or formal language for representing specifications, as well as a systematic methodology for generating specifications. (Howden, 1982)

Interpretive Models

PSL/PSA - Problem State Language/Problem State Analyzer. Developed by the ISDOS project at the University of Michigan. Specialized version of a software engineering data base for storing objects, object properties, and relationships between objects. Employs both the formal language of objects and relations and a graphical notation for displaying relationships. May result in excessive overhead for small projects. (Zelkowitz, Shaw, and Gannon 1979), (Teichroew & Hershey, 1979)

HOS/USE - Higher Order Software Methodology by Higher Order Software Inc. of Cambridge, Mass. This is one of the most complete and perhaps advanced of any of the ADP user-oriented systems. It includes the following automated tools: (1) an interactive graphics editor for entering and editing HOS CONTROL MAPS (2) a mathematically based specification language called AXES (3) Libraries of generalized data types, primitives, defined structures, and interface specifications (4) an Automatic Analyzer used to detect specification errors, (5) a Resource Allocation Tool for converting the analyzed specifications from the automatic analyzer directly into executable code, and (6) an Interactive Simulator to test partially implemented structures and perform prototyping. The development tool that links these parts together and implements the methodology is called USE.IT.

In HOS, the user presents an analyst with a statement of system requirements. The analyst acts as a consultant to the user by translating his requirements interactively into functional specifications in the form of a hierarchical control map using a simple graphical-editor interface. Each node of the hierarchical diagram is specified in terms of its inputs, process, outputs, and control type. The analyzer then automatically

converts the top-down hierarchical control structures from the graphical editor into a formal specification language called AXES. One of the most powerful features of this system is its ability to synthesize complex applications from libraries of reusable primitative operations that can be defined by the user. HOS is also compatible with SADT.

SREM - System Requirements Engineering Methodology, a TRW requirements-analysis system for real-time systems. Uses "stimulus response" diagrams. Consists of a requirements-statement language (RSL) to specify the relationships among the objects of a system. This system is sometimes called the Software Requirements Methodology (SRM). (Zelkowitz, et al. 1979)

SDS - Software Development System by TRW. SDS is one of the more advanced methodologies incorporating many different tools, including: Software Requirements Engineering Methodology (SREM), Requirements Statement Language (RSL), Requirements Engineering and Validation System (REVS), and Program Design Language 2 (PDL2).

SADT - Structured Analysis & Design Technique produced by Softech Inc. Requirements representation diagrams (graphical requirements language) uses a manual graphics system for design and analysis. Uses special forms for denoting system components, their relationships with other components, and input and output data. (Ross & Schoman, 1977) (Zelkowitz, et al. 1979)

BPT - Build Program Technique by John Rice & Olson Research Associates, BPT is the means by which the programmer/analyst can build an Automatic Software Generation System (ASGS). Specifications are communicated to the build programs by a Requirements Specification Language (RSL) tailored to the specific class of user. The RSL statements are then decoded and transferred

to the "build programs module" by the RSL analyzer. The RSL analyzer uses various skeletal software construction tailored for the users and stored in an Intermediate File (IMF).(Rice, 1981)

Prototype Systems/Languages

ACT/1 - An IBM compatible prototyping system developed by Bailey & Rose of Toronto, Ontario. The essence of the methodology is that the system designer or architect develops a view of the system based on the system's external description or appearance, a view that is both understandable and acceptable to the users. The first set of iterations makes use of "scenarios" constructed from sequences of side play screens. This iteration process leads to agreement on such key matters as screen-flow-sequences, screen content, and whether the application is to be menu-driven or forms-driven, question and answer, etc. The system specifications are represented by a series of machine-implemented application scenarios, rather than functional flowcharts or application/structure diagrams. The developer concentrates on the design and implementation of program transaction modules that process data between screens. (Mason, Carey 1983)

ADMINIS/11 - By Adminis Corp. for PDP/11. has graphics generator, report generator, and application generator. Has on-line capability using its own file structure, and is suitable for end-users.

DATA ANALYZER - By Programming Products for IBM 370 systems. Has a report generator, graphics generator, query language, and is on-line. Suitable for end-users and has its own file structure.

FOCU - By Info. Builders for IBM 370. Has an application generator, report generator, graphics generator, query language, very high-level programming facilities, and is on-line. Suitable for end-users and has its own file structure.

NOMAD -	By National C.S.S. for N.C.S.S. Has a relational data base with a report generator, graphics generator, query language, very high-level programming facilities, and is on-line. Suitable for end-users.
USER/11 -	By Northcounty for PDP/11. Has an application generator, a report generator, graphics generator, query language, very high-level programming facilities, and is on-line. Suitable for end-users and has its own file structure.

Design Methods

PDL -	Program Design Language (META-languages for precisely defining program modules, including all data structures and operations on the data).
STRUCTURED DESIGN -	Manual System for creating a detailed design, consisting of: subsystem, process, activity, and module. Includes development of hand-drawn baseline diagram. (Zelkowitz, et al., 1979)
JDM -	Jackson Design Methodology - based on the principle that, at the specification stage of a program, the element that exhibits the most structure is that of the data. The Jackson approach is to define the structure of the data and then to derive the program structure from the data structures. Incorporates the techniques of top-down development, structured programming, and structured walkthroughs. Programs are hierarchically structured with the following four basic components: elementary, sequence, selection, and iteration. (King, 1981) (Zelkowitz, et al. 1979)
WARNIER-ORR - DIAGRAMS	Warnier's approach includes three separate techniques known as: Logical Construction of Systems (LCS), Logical Construction of Programs (LCP), and Logical Construction of Execution (LCE). Warnier's techniques have been expanded and marketed in the U.S. by Ken Orr and Associates. (Orr, 1981) (King, 1981)

INFORMATION - ENGINEERING

Developed by INFOCOM, Australia. Data held in computer files or data bases and used to "model" the organization. IE has an initial, data-oriented analysis followed by a procedure-oriented approach.

DATA-ORIENTED

1. Examine the corporate purpose and mission; identify fundamental data:

   a. Current organization and mission
   b. The direction the organization is headed.
   c. The direction the organization should be headed in.

2. Identify data required for specific functional areas.

3. Identify data needed for top management decision making.

4. Data base design to model the organization.

PROCEDURE-ORIENTED

1. Identify decision events which bring about data change.

2. Users develop formal, structured-English procedure specifications. (Finkelstein, 1981)

DATA - DICTIONARIES

Used for recording in a centralized location all decisions related to the structure and implementation of every element, record, and file.

Management Systems/Tools

<u>Manual Tools</u>

Milestones
Gantt Charts
Pert
Critical Path

Automated Management Systems

Estimator - For IBM systems by AGS Management Systems Inc. An estimation tool for determining the time and cost estimates for each phase of a system-development life cycle.

G/C CUE - For H-P 3000, Prime, Dec & Vax Equipment by Gilbert/Commonwealth. Provides estimating, accounting, planning, scheduling, and cost-performance measurement information.

Spectrum-3- For IBM & Apple computers by Spectrum International Inc. A project estimating tool which provided estimated guidelines in terms of person hours, cost, and schedule of the project work.

CSSR - Runs on H-P 3000 Systems, produced by AGS Management Systems, Inc. Monitors and forecasts cost and schedule performance on smaller acquisitions/projects.

PAC II/III - Runs on IBM, DEC VAX, and System 10-20 computers, by AGS Management Systems, Inc. Provides project management for large projects. Combines cost,time and resource factors to forecast when each project activity will be completed, how and at what cost. Calculates critical paths, resource bottlenecks, and keeps detailed cost/accounting information.

PC/70 - Runs on IBM and HP 3000 Systems, by AGS Management Systems, Inc. Designed to forecast schedules, cost-and workloads; pinpoint trouble spots; simulate schedules, measure performance and progress.

N5500 Project Planning and Control System - Runs on IBM, DEC, Burroughs, HP 3000, Honeywell, CDC, UNIVAC, Prime, WANG. Built by Nicholas & Company, determines project trends and predicts completion dates & costs.

Prompt Aid 1 (for estimator) - Runs on an Intertec Superbrain, produced by Simpact Systems. Assists users in determining the expected cost and effort of a computer development project.

APPENDIX B

RS Development in DOD

Richhart (1983) gives the following description of RS development and use in the DOD:

The U.S. Military has recognized the need to increase the amount of attention given to identifying user needs and to developing requirements specifications. The U.S. Navy has implemented DOD guidance for developing an ADP system through the use of NAVDAC PUB 24.1 and 24.2. The U.S. Air Force has implemented this guidance in Air Force regulation AFR 300-12 and AFR 300-15.

Prior to the development of System Specifications (SS) in the definition phase, both the Navy and the Air Force require a mission-analysis phase and a concept-development phase. The mission-analysis phase is intended as a means for extracting and identifying the essential needs of the user, recognizing that the user may not know exactly what the problem is or how ADP resources might best be able to help. In the Navy, this phase is accompanied by the development of a Mission Element Need Statement (MENS) and a statement of General Functional Requirements (GFR). In the Air Force, a feasibility study is performed in conjunction with the identification of the existing functional baseline.

Both services follow their initial investigation with a conceptual phase to develop the initial Concept of Operations (CONOPS) and an economic analysis (USAF) or System Decision Paper (SDP) (USN) to determine whether further development effort is justified. If a decision is made to proceed, a detailed Functional Description (FD) is prepared describing the functions to be developed or integrated with a new system. In the Navy, a Plan of Action and Milestones (POA & M) document is prepared. The Air Force equivalent of the POA & M is the Data Project Plan (DPP). When all of this documentation is ready, and the users have had a chance to review it, a System Requirements Review (SRR) is held with key representatives from all of the user and ADP offices attending. The SRR is a rather painstaking, drawn out review that frequently proceeds paragraph by paragraph through the FD and milestones. It is not at all unusual for the users to ask so many questions or make so many corrections that a follow-up SRR is required to resolve the differences. When the SRR is finished and the corrected FD and milestones are approved by the user, they form a new functional baseline.

In the system-design phase, the detailed System Specifications (SS) are developed along with a Data Requirements Document (DRD), and an Interface Control Document (ICD). These documents are reviewed by the users and approved in a System Design Review (SDR) which is very similar to the SRR. The SDR establishes a new design baseline for the hardware, communications, and software (called an "allocated baseline" in the USAF).

The SDR is followed by a more detailed subsystem-design and data-base specification that is finalized in a Preliminary Design Review (PDR). In smaller systems, the PDR and SDR are often combined into one larger SDR. Each program then goes through a design phase and the development of detailed Program Specifications (PS) which are approved in small Critical Design Reviews (CDR) between the responsible programmer, his/her manager, and the individual user who will eventually use the program.

All of the preceeding work is done before the System Development Phase gets the developer involved with actual programming of the individual modules. While this looks okay on paper, obviously there is a lot of overhead involved with this type of specification and design. To shorten the process so that small projects are not hampered, thresholds are used to select the projects that must follow the full development schedule. Even when the size of a project should require it to follow these guidelines, it is often excused because of an operational necessity to meet operational dates.

APPENDIX C

Costing Matrices

The matrices in Figures C1 - C7 are the costing matrices described in the section "Method of Approach: The Participant's Task."

50.00,50.00,50.00,2.50,50.00,50.00,2.50,50.00,5.00,25.00,50.00
50.00,5.00,50.00,5.00,50.00,7.50,50.00,7.50,5.00,7.50,5.00
50.00,50.00,5.00,7.50,50.00,50.00,5.00,7.50,50.00,50.00,5.00
7.50,50.00,50.00,5.00,7.50,50.00,50.00,50.00,7.50,50.00,50.00

Note: This is a sequential 4x11 array beginning with i=1, j=1 and ending with i=14, j=4. The index j takes on the maximum value of 2, 3, or 4 depending on the value of i. See Figure C3 for the full run of i and j.

Figure C1. $\overline{C_{ij}}$

.05,.01,.001,.01,.05,.05,.03,.05,.10,.10,.05
.05,.05,.05,.05,.05,.01,.05,.01,.05,.01,.05
.10,.001,.05,.01,.10,.001,.05,.01,.10,.001,.05
.01,.10,.001,.05,.01,.10,.001,.05,.01,.10,.001

Note: This is a sequential 4x11 array beginning with i=1, j=1 and ending with i=14, j=4. The index j takes on the maximum value of 2, 3, or 4 depending on the value of i. See Figure C3 for the full run of i and j.

Figure C2. PE_{ij}

i=	j=	k=1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	0.00	0.00	0.00	0.90	0.00	0.00	0.00	0.01	0.00	0.04	0.00	0.00	0.00
	2	0.16	0.00	0.53	0.80	0.01	0.00	0.00	0.82	0.00	0.87	0.00	0.00	0.00
	3	0.84	0.00	0.17	0.48	0.32	0.00	0.00	0.65	0.00	0.99	0.00	0.00	0.00
	4	0.11	0.00	0.73	0.39	0.83	0.00	0.00	0.42	0.00	0.07	0.00	0.00	0.00
2	1	0.00	0.66	0.31	0.93	0.78	0.00	0.33	0.00	0.96	0.00	0.00	0.00	0.00
	2	0.00	0.83	0.31	0.41	0.71	0.00	0.52	0.00	0.73	0.00	0.00	0.00	0.00
	3	0.00	0.74	0.88	0.63	0.83	0.00	0.32	0.00	0.39	0.00	0.00	0.00	0.00
3	1	0.50	0.00	0.47	0.31	0.60	0.00	0.00	0.00	0.00	0.00	0.85	0.68	0.45
	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.62	0.65	0.28
	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.86	0.63	0.04
4	1	0.55	0.00	0.05	0.77	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.54
	2	0.00	0.00	0.00	0.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.53
	3	0.00	0.00	0.00	0.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.35
5	1	0.32	0.00	0.78	0.83	0.91	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.91
	2	0.90	0.00	0.00	0.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.33	0.80
6	1	0.84	0.00	0.81	0.37	0.89	0.00	0.00	0.00	0.00	0.00	0.02	0.07	0.28
	2	0.00	0.00	0.04	0.00	0.74	0.00	0.00	0.00	0.00	0.00	0.11	0.98	0.93
7	1	0.71	0.00	0.89	1.00	0.95	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.69
	2	0.00	0.00	0.90	0.90	0.63	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.55
8	1	0.66	0.00	0.02	0.17	0.84	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.57
	2	0.00	0.00	0.00	0.00	0.82	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.77
9	1	0.28	0.00	0.74	0.91	0.78	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.49
	2	0.94	0.00	0.22	0.85	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.92
	3	0.62	0.00	0.44	0.06	0.42	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.96
10	1	0.03	0.00	0.47	0.61	0.39	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.86
	2	0.00	0.00	0.00	0.71	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.49
	3	0.52	0.00	0.75	0.79	0.96	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.69
	4	0.50	0.00	0.77	0.12	0.81	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75
11	1	0.20	0.00	0.47	0.99	0.71	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.33
	2	0.00	0.00	0.00	0.00	0.59	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.63
	3	0.46	0.00	0.02	0.02	0.94	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.47
	4	0.35	0.00	0.80	0.70	0.96	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.53
12	1	0.48	0.00	0.16	0.62	0.28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10
	2	0.00	0.00	0.00	0.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.49
	3	0.36	0.00	0.78	0.42	0.49	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.18
	4	0.63	0.00	0.17	0.31	0.39	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.48
13	1	0.42	0.00	0.18	0.30	0.16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.28
	2	0.00	0.00	0.00	0.00	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02
	3	0.87	0.00	1.00	0.20	0.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.37
	4	0.49	0.00	0.59	0.13	0.51	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.88
14	1	0.71	0.00	0.36	0.74	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.53
	2	0.00	0.00	0.00	0.36	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.35
	3	0.90	0.00	0.23	0.32	0.84	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.13
	4	0.19	0.00	0.00	0.29	0.74	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.81

Figure C3. PD_{kij} 1

i=	j=	k=1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	0.25	0.00	0.16	0.73	1.00	0.00	0.00	0.37	0.00	0.25	0.00	0.00	0.00
	2	0.17	0.00	0.76	0.04	0.44	0.00	0.00	0.27	0.00	0.68	0.00	0.00	0.00
	3	0.64	0.00	0.76	0.77	0.82	0.00	0.00	0.98	0.00	0.51	0.00	0.00	0.00
	4	0.22	0.00	0.70	0.22	0.03	0.00	0.00	0.19	0.00	0.93	0.00	0.00	0.00
2	1	0.00	0.83	0.62	0.31	0.22	0.00	0.59	0.00	0.53	0.00	0.00	0.00	0.00
	2	0.00	0.85	0.34	0.39	0.25	0.00	0.97	0.00	0.57	0.00	0.00	0.00	0.00
	3	0.00	0.70	0.09	0.20	0.42	0.00	0.70	0.00	0.43	0.00	0.00	0.00	0.00
3	1	0.34	0.00	0.15	0.79	0.43	0.00	0.00	0.00	0.00	0.00	0.43	0.74	0.55
	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.62	0.76	0.93
	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.78	0.29	0.75
4	1	0.90	0.00	0.68	0.98	0.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.39
	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.08
	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.97
5	1	0.06	0.00	0.64	0.33	0.21	0.00	0.00	0.00	0.00	0.00	0.00	0.31	0.98
	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.43
6	1	0.23	0.00	0.58	0.35	0.92	0.00	0.00	0.00	0.00	0.00	0.39	0.04	0.70
	2	0.00	0.00	0.85	0.00	0.85	0.00	0.00	0.00	0.00	0.00	0.41	0.85	0.41
7	1	0.84	0.00	0.29	0.23	0.74	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.39
	2	0.00	0.00	0.00	0.00	0.62	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.27
8	1	0.02	0.00	0.66	0.79	0.77	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.49
	2	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.69
9	1	0.02	0.00	0.95	0.48	0.35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.76
	2	0.43	0.00	0.77	0.75	0.56	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.59
	3	0.46	0.00	0.50	0.83	0.47	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.35
10	1	0.87	0.00	0.03	0.35	0.86	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.98
	2	0.00	0.00	0.00	0.16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.12
	3	0.30	0.00	0.67	0.34	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95
	4	0.65	0.00	0.32	0.10	0.76	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.60
11	1	0.82	0.00	0.49	0.55	0.93	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.63
	2	0.00	0.00	0.00	0.00	0.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.70
	3	0.60	0.00	0.27	0.24	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.83
	4	0.06	0.00	0.87	0.69	0.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.62
12	1	0.93	0.00	0.10	0.31	0.52	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.29
	2	0.80	0.00	0.00	0.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75
	3	0.72	0.00	0.10	0.67	0.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.78
	4	0.95	0.00	0.10	0.62	0.43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.98
13	1	0.99	0.00	0.12	0.80	0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.28
	2	0.00	0.00	0.00	0.00	0.88	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.78
	3	0.80	0.00	0.78	0.44	0.61	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.74
	4	0.94	0.00	0.00	0.53	0.16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.19
14	1	0.73	0.00	0.63	0.21	0.60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.70
	2	0.00	0.00	0.00	0.79	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.41
	3	0.32	0.00	0.24	0.57	0.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.64
	4	0.12	0.00	0.96	0.71	0.63	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.41

Figure C4. $PD_{kij}{}^2$

i=	j=	k= 1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	0.88	0.00	0.28	0.78	0.15	0.00	0.00	0.93	0.00	0.19	0.00	0.00	0.00
	2	0.81	0.00	0.15	0.59	0.16	0.00	0.00	0.64	0.00	0.43	0.00	0.00	0.00
	3	0.77	0.00	0.75	0.61	0.87	0.00	0.00	0.77	0.00	0.79	0.00	0.00	0.00
	4	0.78	0.00	0.59	0.54	0.96	0.00	0.00	0.86	0.00	0.57	0.00	0.00	0.00
2	1	0.00	0.69	0.97	0.62	0.97	0.00	0.26	0.00	0.83	0.00	0.00	0.00	0.00
	2	0.00	0.66	0.50	0.06	0.85	0.00	0.56	0.00	0.76	0.00	0.00	0.00	0.00
	3	0.00	0.48	0.07	0.09	0.90	0.00	0.63	0.00	0.69	0.00	0.00	0.00	0.00
3	1	0.44	0.00	0.45	0.69	0.11	0.00	0.00	0.00	0.00	0.00	0.46	0.75	0.33
	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.25	0.51	0.79
	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.15	0.73	0.09
4	1	0.94	0.00	0.84	0.59	0.92	0.00	0.80	0.00	0.00	0.00	0.00	0.00	0.26
	2	0.00	0.00	0.00	0.00	0.00	0.00	0.80	0.00	0.00	0.00	0.00	0.00	0.23
	3	0.00	0.00	0.00	0.00	0.00	0.00	0.90	0.00	0.00	0.00	0.00	0.00	0.09
5	1	0.47	0.00	1.00	0.72	0.37	0.00	0.00	0.00	0.00	0.00	0.00	0.72	0.96
	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.33	0.35
6	1	0.06	0.00	0.26	0.02	0.74	0.00	0.80	0.80	0.00	0.00	0.31	0.16	0.19
	2	0.00	0.00	0.70	0.00	0.48	0.00	0.70	0.70	0.00	0.00	0.62	0.37	0.63
7	1	0.47	0.00	0.16	0.74	0.55	0.00	0.60	0.50	0.00	0.00	0.00	0.00	0.28
	2	0.00	0.00	0.00	0.00	0.68	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.62
8	1	0.57	0.00	0.82	0.81	0.52	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.80
	2	0.00	0.00	0.00	0.00	0.12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.51
9	1	0.93	0.00	0.03	0.79	0.43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.47
	2	0.97	0.00	0.59	0.79	0.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.28
	3	0.13	0.00	0.25	0.34	0.73	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.36
10	1	0.56	0.00	0.12	0.66	0.89	0.00	0.60	0.40	0.00	0.00	0.00	0.00	0.42
	2	0.00	0.00	0.00	0.51	0.00	0.00	0.50	0.40	0.00	0.00	0.00	0.00	0.26
	3	0.46	0.00	0.41	0.86	0.44	0.00	0.70	0.50	0.00	0.00	0.00	0.00	0.89
	4	0.40	0.00	0.39	0.75	1.00	0.00	0.80	0.60	0.00	0.00	0.00	0.00	0.18
11	1	0.13	0.00	0.15	0.71	0.94	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.25
	2	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.60	0.00	0.00	0.00	0.00	0.90
	3	0.16	0.00	0.50	0.40	0.19	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.55
	4	0.59	0.00	0.58	0.17	0.78	0.00	0.00	0.30	0.00	0.00	0.00	0.00	0.21
12	1	0.21	0.00	0.39	0.42	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.16
	2	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.87
	3	0.78	0.00	0.85	0.10	0.89	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.48
	4	0.86	0.00	0.84	0.29	0.12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.15
13	1	0.79	0.00	0.42	0.41	0.71	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.60
	2	0.00	0.00	0.00	0.00	0.17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.64
	3	0.31	0.00	0.13	0.99	0.77	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.71
	4	0.34	0.00	0.62	0.64	0.29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.96
14	1	0.13	0.00	0.10	0.46	0.84	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.92
	2	0.00	0.00	0.00	0.98	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.59
	3	0.77	0.00	0.30	0.81	0.18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.80
	4	0.22	0.00	0.09	0.59	0.69	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.88

Figure C5. PD_{kij}^3

i=	j= \ k=	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	0.78	0.00	0.03	0.13	0.49	0.00	0.00	0.79	0.00	0.35	0.00	0.00	0.00
	2	0.00	0.00	0.85	0.10	0.91	0.00	0.00	0.60	0.00	0.40	0.00	0.00	0.00
	3	0.04	0.00	0.58	0.18	0.80	0.00	0.00	0.20	0.00	0.01	0.00	0.00	0.00
	4	0.26	0.00	0.48	0.54	0.94	0.00	0.00	0.76	0.00	0.05	0.00	0.00	0.00
2	1	0.00	0.50	0.53	0.73	0.57	0.00	0.86	0.00	0.07	0.00	0.00	0.00	0.00
	2	0.00	0.68	0.43	0.46	0.90	0.00	0.26	0.00	0.45	0.00	0.00	0.00	0.00
	3	0.00	0.35	0.06	0.22	0.74	0.00	0.50	0.00	0.33	0.00	0.00	0.00	0.00
3	1	0.51	0.00	0.06	0.80	0.27	0.00	0.00	0.00	0.00	0.00	0.44	0.21	0.30
	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.91	0.72	0.16
	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.52	0.66	0.23
4	1	0.46	0.00	0.72	0.16	0.51	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.89
	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.99
	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.80
5	1	0.73	0.00	0.64	0.10	0.16	0.00	0.50	0.50	0.50	0.50	0.00	0.06	0.91
	2	0.00	0.00	0.00	0.00	0.00	0.00	0.60	0.60	0.60	0.60	0.00	0.90	0.75
6	1	0.35	0.00	0.59	0.17	0.08	0.00	0.50	0.50	0.50	0.50	0.94	0.64	0.84
	2	0.00	0.00	0.31	0.00	0.26	0.00	0.60	0.60	0.60	0.60	0.89	0.39	0.66
7	1	0.05	0.00	0.47	0.35	0.42	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.65
	2	0.00	0.00	0.00	0.00	0.32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.78
8	1	0.82	0.00	0.89	0.93	0.55	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95
	2	0.00	0.00	0.00	0.00	0.73	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.81
9	1	0.34	0.00	0.73	0.28	0.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.26
	2	0.34	0.00	0.71	0.20	0.86	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.35
	3	0.35	0.00	0.95	0.58	0.91	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.28
10	1	0.47	0.00	0.29	0.55	0.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.22
	2	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.23
	3	0.11	0.00	0.60	0.62	0.29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.17
	4	0.45	0.00	0.15	0.90	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06
11	1	0.20	0.00	0.65	0.06	0.42	0.00	0.50	0.50	0.50	0.50	0.00	0.00	0.60
	2	0.00	0.00	0.00	0.00	0.50	0.00	0.60	0.60	0.60	0.60	0.00	0.00	0.96
	3	0.42	0.00	0.26	0.47	0.66	0.00	0.70	0.70	0.70	0.70	0.00	0.00	0.92
	4	0.48	0.00	0.42	0.52	0.32	0.00	0.60	0.60	0.60	0.60	0.00	0.00	0.81
12	1	0.39	0.00	0.48	0.37	0.93	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.21
	2	0.00	0.00	0.00	0.88	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.40
	3	0.51	0.00	0.43	0.00	0.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75
	4	0.34	0.00	0.31	0.81	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.09
13	1	0.48	0.00	0.60	0.50	0.61	0.00	0.50	0.50	0.50	0.50	0.00	0.00	0.68
	2	0.00	0.00	0.00	0.00	0.61	0.00	0.60	0.60	0.60	0.60	0.00	0.00	0.57
	3	0.59	0.00	0.57	0.59	0.61	0.00	0.70	0.70	0.70	0.70	0.00	0.00	0.81
	4	0.39	0.00	0.53	0.71	0.50	0.00	0.60	0.60	0.60	0.60	0.00	0.00	0.59
14	1	0.02	0.00	0.80	0.60	0.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95
	2	0.00	0.00	0.00	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.28
	3	0.43	0.00	0.09	0.69	0.31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.69
	4	0.32	0.00	0.71	0.34	0.68	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

Figure C6. PD_{kij}^{4}

k=	1	2	3	4	5	6	7	8	9	10	11	12	13
	200.,	200.,	100.,	250.,	250.,	50.,	100.,	100.,	100.,	100.,	500.,	1000.,	2000.

Figure C7. CT_k

APPENDIX D

Newspaper Advertisement

Newspaper Ad placed in the Washington Post on May 1, 1983, May 8, 1983, and June 12, 1983:

MANAGEMENT: A firm in Vienna is seeking the help of 36 managers with inventory control experience to evaluate a computer system. This work is sponsored by the Department of Navy. The computer evaluation will take approximately 4 1/2 hours (1 day only). Qualifications: Degree in Business Management or 4 years management experience, including inventory control. $12.00/hr. Call Susan for details. 9 a.m. to 1 p.m. Mon-Fri. 938-1603.

APPENDIX E

Forms Used in the Experiment

Participant No. ______________

Date: ______________

PARTICIPANT BIOGRAPHICAL FORM

Name: ____

Age: ____________________ Sex: ____________

Education: High School Graduate: Yes __________ No __________

Years of higher education ______________ Degree: ______________

Major: ____________________________ Minor: ______________

Management Experience:

____

____

____

____

Inventory Control Systems:

(1) ____

(2) ____

(3) ____

(4) ____

(5) ____

Please make any additional comments you believe best describes your professional experience.

Higher Level Language Programming Experience

Please check all languages in which you have programmed and list your best estimate of the number of programs coded in that language.

	TYPE	NUMBER OF PROGRAMS CODED
______	BASIC	
______	FORTRAN	
______	COBOL	
______	PL-1	
______	ALGOL	
______	PASCAL	
	______	
	______	

Enter your best estimate of the total number of programs you have coded (any language, any computer). ____________

- -

Data Processing Experience

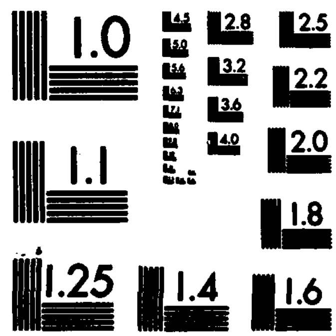
Enter the number of months of full-time experience you have in the following:

Number of Months	
______	Data Entry
______	Production Control
______	Operations
______	Applications Programming
______	System Programming
______	System Analysis
______	Data Base Administration
______	Data Communications
______	Other(s)

Please give the number of year's you have worked in the computer field. ____________

AD-A140 149 EMPIRICAL INVESTIGATION OF AIDS FOR NON-PROGRAMMING USERS IN DEVELOPING C..(U) PERFORMANCE MEASUREMENT ASSOCIATES INC VIENNA VA E M CONNELLY MAR 84 2/2

UNCLASSIFIED PMA-84-822 N00014-82-C-0499 F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Enter your best estimate of the number of programs for which you have coded the Job Control Language (JCL) necessary for testing or production runs. ______________

- -

Please list up to 5 computer/operating systems on which you have worked which you believe best represents your experience. Examples: IBM 370/155 OS, PDP-11 RT-11.

- -

Specifications of Data Base Systems

Please list the number of Data Base Systems including inventory systems you have used: For each indicate if you are an user, specific or designer.

Data Base System	User	Specific	Designer

Please enter the number of years experience with Data Base Systems.

____

CONTRACT TO ACT AS A RESEARCH PARTICIPANT
FOR
THE DEPARTMENT OF THE NAVY

Name of Participant: ______________________ Date: __________

Address: ____
Street City State Zip Code

1. I authorize Mr. Edward M. Connelly, who is the Principal Investigator for this project, or his representative, to collect and analyze my solutions to test problems.

2. This project has been explained to me by ______________________
(Print)

   It has been pointed out to me that my solutions will not be made known to any individual, other than appropriate members of the research team, or for any purpose other than the data analysis to be performed by the research team. No information concerning my performance on the experimental task will be disclosed to anyone other than the research team without my written permission.

3. I understand that there are no special risks of any kind associated with my participation in this study.

4. I understand that the project may further the understanding of how various aids can assist a programmer or other person to specify a computer program.

5. I understand that Mr. Edward M. Connelly or Joanne C. Connelly will answer any questions that I may have about this project.

6. I understand that by signing this form, I have waived none of my legal rights that may be associated with liability for negligence on the part of Performance Measurement Associates, Inc.

7. I state that I am 18 years of age or older.

____
(Participant Signature)

____
(Administrator)

REFERENCES

Anderson, V. L., and McLean, R. A. Design of experiments. New York: Marcel Denkker, Inc., 1974.

Bearley, W. L., and Wood, M. R. "Systems Planning," Computerworld, Framingham, MA: CW Cummunications Inc., 1980.

Boehm, B. W. Software engineering. IEEE Transactions on Computers. Vol. C-25, No. 12, pp. 1226-41, December 1976.

Boehm, B. W. Software engineering economics. Englewood Cliffs: Prentice-Hall Inc., 1981.

Boole, G. An investigation of the laws of thought. New York: Dover Publication, Inc., 1854.

Finkelstein, C. "Information Engineering", Computerworld, Framingham, MA: CW Communications Inc., 11 May, 25 May, 1 Jun, 8 Jun, 15 Jun 1981.

Howden, W. E. "Contemporary Software Development Environments", Communications of the ACM, May 1982.

Jackson, M. A. Principles of program design. Academic Press, New York, 1975.

King, D. "Current Methodologies in Structured Design", Computerworld, Framingham, MA: CW Cummunications Inc., 1981.

Martin, J. Application development without programmers. Englewood Cliffs: Prentice-Hall Inc., 1982.

Mason, R. E. A., and Carey, T. T. "Prototyping Interactive Information Systems", Communications of the ACM, 1983.

Miller, L. A. Behavioral studies of the programming process. Final Report for ONR, Work Unit NR 197-020, Code 455 October 1978.

Nassi, I., and Schneiderman, B. Flow chart technique for structured programming. ACM Sigplan Notice 8, 1973.

REFERENCES (CONTD)

Nolan, D. "Natural Project Control", Computerworld, Framingham, MA: CW Communications Inc., 1981.

Orr, L. Structured requirements definition. Topeka: Ken Orr & Associates, 1981.

Ramamoorthy, C. V., and So, H. H. Software requirements and specifications: status and perspectives. Presented at the IEEE Computer Society's Second International Computer Software and Applications Conference, Chicago, Illinois, 1978.

Rice, J. G. Build program technique: a practical approach for the development of automatic software generation systems. New York: John Wiley & Sons, 1981.

Richhart, K. M. "Tools for developing ADP requirements specifications." Vienna, VA: Performance Measurement Associates, Inc. 1983.

Ross, D. T., and Schoman, K. E., Jr. Structured analysis for requirements definition. IEEE Transactions on Software Engineering, 1977.

Ross, J. E. Modern management and information systems. Reston: Reston Publishing Company, 1976.

Sokal, R. R., and Rohlf, F. J. Biometry. San Francisco: W. H. Freeman & Company, 1969.

Teichroew, D., and Hershey, E. A. III. PSL/PSA: a computer-aided technique for structured documentation and analysis of information processing systems. Presented at the IEEE Computer Society's Third International Computer Software & Applications Conference, Chicago, Illinois 1979.

Warnier, J. D. Logical construction of systems. New York: Van Nostrand Reinhold, 1981.

DISTRIBUTION LIST

OSD

CAPT Paul R. Chatelier
Office of the Deputy Under
Secretary of Defense
OUSDRE (E&LS)
Pentagon, Room 3D129
Washington, D.C. 20301

Dr. Dennis Leedom
Office of the Deputy Under
Secretary of Defense (C^3I)
Pentagon
Washington, D.C. 20301

Department of the Navy

Engineering Psychology Programs
Office of Naval Research
Code 442EP
800 N. Quincy Street
Arlington, VA 22217 (3 cys.)

Physiology Program
Office of Naval Research
Code 441NP
800 North Quincy Street
Arlington, VA 22217

Dr. Andrew Rechnitzer
Office of the Chief of Naval
Operations, OP952F
Naval Oceanography Division
Washington, D.C. 20350

Manpower, Personnel & Training
Programs
Code 270
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Department of the Navy

Information Sciences Division
Code 433
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Special Assistant for Marine
Corps Matters
Code 100M
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

CDR James Offutt, Officer-in-Charge
ONR Detachment
1030 East Green Street
Pasadena, CA 91106

Director
Naval Research Laboratory
Technical Information Division
Code 2627
Washington, D.C. 20375

Dr. J. S. Lawson
Naval Electronic Systems Command
NELEX-06T
Washington, D.C. 20360

Naval Training Equipment Center
ATTN: Technical Library
Orlando, FL. 32813

Dr. Robert G. Smith
Office of the Chief of Naval
Operations, OP987H
Personnel Logistics Plans
Washington, D.C. 20350

Department of the Navy

Dr. Michael Melich
Communications Sciences Division
Code 7500
Naval Research Laboratory
Washington, D.C. 20375

Combat Control Systems
Department
Code 35
Naval Underwater Systems
Center
Newport, RI 02840

Human Factors Department
Code N-71
Naval Training Equipment
Center
Orlando, FL 32813

Dr. Alfred F. Smode
Training Analysis and
Evaluation Group
Naval Training & Equip-
ment Center
Orlando, FL 32813

CDR Norman E. Lane
Code N-7A
Naval Training Equipment
Center
Orlando, FL 32813

Dean of Research Administration
Naval Postgraduate School
Monterey, CA 93940

Human Factors Engineering
Code 8231
Naval Ocean Systems Center
San Diego, CA 92152

Department of the Navy

Dr. A. L. Slafkosky
Scientific Advisor
Commandant of the Marine Corps
Code RD-1
Washington, D.C. 20380

Dr. L. Chmura
Naval Research Laboratory
Code 7592
Computer Sciences & Systems
Washington, D.C. 20375

Office of the Chief of Naval
Operations (OP-115)
Washington, D.C. 20350

CDR C. Hutchins
Code 55
Naval Postgraduate School
Monterey, CA 93940

Human Factors Technology Administrator
Office of Naval Technology
Code MAT 0722
800 N. Quincy Street
Arlington, VA 22217

Commander
Naval Air Systems Command
Human Factors Programs
NAVAIR 334A
Washington, D.C. 20361

Commander
Naval Air Systems Command
Crew Station Design
NAVAIR 5313
Washington, D.C. 20361

Department of the Navy

Mr. Philip Andrews
Naval Sea Systems Command
NAVSEA 61R
Washington, D.C. 20362

Commander
Naval Electronics Systems Command
Human Factors Engineering Branch
Code 81323
Washington, D.C. 20360

Larry Olmstead
Naval Surface Weapons Center
NSWC/DL
Code N-32
Dahlgren, VA 22448

Dr. George Moeller
Human Factors Engineering Branch
Submarine Medical Research Lab
Naval Submarine Base
Groton, CT 06340

Navy Personnel Reseearch & Development Center
Planning & Appraisal Division
San Diego, CA 92152

Dr. Robert Blanchard
Navy Personnel Research & Development Center
Command and Support Systems
San Diego, CA 92152

CDR J. Funaro
Human Factors Engineering Division
Naval Air Development Center
Warminster, PA 18974

Department of the Navy

Mr. Stephen Merriman
Human Factors Engineering Division
Naval Air Development Center
Warminster, PA 18974

Mr. Jeffrey Grossman
Human Factors Branch
Code 3152
Naval Weapons Center
China Lake, CA 93555

Human Factors Engineering Branch
Code 4023
Pacific Missile Test Center
Point Mugu, CA 93042

Dean of the Academic Departments
U.S. Naval Academy
Annapolis, MD 21402

Dr. W. Moroney
Human Factors Section
Systems Engineering Test Directorate
U.S. Naval Air Test Center
Patuxent River, MD 20670

Department of the Army

Dr. Edgar M. Johnson
Technical Director
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Technical Director
U.S. Army Human Engineering Labs
Aberdeen Proving Ground, MD 21005

Department of the Army

Director, Organizations and System Research Lab
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Mr. J. Barber
HQS, Department of the Army
DAPE-MBR
Washington, D.C. 20310

Department of the Air Force

U.S. Air Force Office of Scientific Research
Life Sciences Directorate, NL
Bolling Air Force Base
Washington, D.C. 20332

AFHRL/LRS TDC
Attn: Susan Ewing
Wright-Patterson AFB, OH 45433

Chief, Systems Engineering Branch
Human Engineering Division
USAF AMRL/HES
Wright-Patterson AFB, OH 45433

Dr. Earl Alluisi
Chief Scientist
AFHRL/CCN
Brooks Air Force Base, TX 78235

Foreign Addresses

Dr. Kenneth Gardner
Applied Psychology Unit
Admiralty Marine Tech. Estab.
Teddington, Middlesex TW11 OLN
England

Foreign Addresses

Human Factors
P.O. Box 1085
Station "B"
Rexdale, Ontario
M9V 2B3
Canada

Dr. A. D. Baddeley
Director, Applied Psychology Unit
Medical Research Council
15 Chaucer Road
Cambridge, CB2 2EF England

Other Government Agencies

Defense Technical Information Center
Cameron Station, Bldg. 5
Alexandria, VA 22314 (12 copies)

Dr. Clinton Kelly
Defense Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209

Dr. M. D. Montemerlo
Human Factors & Simulation Technology, RTE-6
NASA HQS
Washington, D.C. 20546

Other Organizations

Ms. Denise Benel
Essex Corporation
333 N. Fairfax Street
Alexandria, VA 22314

Other Organizations

Dr. H. McI. Parsons
Essex Corporation
333 N. Fairfax
Alexandria, VA 22314

Dr. Jesse Orlansky
Institute for Defense Analyses
1801 N. Beauregard Street
Alexandria, VA 22311

Dr. J. O. Chinnis, Jr.
Decision Science Consortium, Inc.
7700 Leesburg Pike
Suite 421
Falls Church, VA 22043

Dr. Paul E. Lehner
PAR Technology Corp.
P.O. Box 2005
Reston, VA 22090

NAS- National Research Council
(COHF)
2101 Constitution Avenue, N.W.
Washington, D.C. 20418

Dr. Deborah Boehm-Davis
General Electric Company
Information & Data Systems
1755 Jefferson Davis Highway
Arlington, VA 22202

Dr. James H. Howard, Jr.
Department of Psychology
Catholic University
Washington, D.C. 20064

Dr. Edward R. Jones
Chief, Human Factors Engineering
McDonnell-Douglas Astronautics Co.
St. Louis Division
Box 516
St. Louis, MO 63166

Other Organizations

National Security Agency
ATTN: N-32, Marie Goldberg
9800 Savage Road
Ft. Meade, MD 20722

Dr. Marvin Cohen
Decision Science Consortium, Inc.
Suite 721
7700 Leesburg Pike
Falls Church, VA 22043

Dr. Richard Pew
Bolt Beranek & Newman, Inc.
50 Moulton Street
Cambridge, MA 02238

Dr. David J. Getty
Bolt Beranek & Newman, Inc.
50 Moulton Street
Cambridge, MA 02238

Dr. Alan Morse
Intelligent Software Systems Inc.
160 Old Farm Road
Amherst, MA 01002

END

FILMED

5-84

DTIC